ABSTRACT

Title of thesis: DEVELOPMENT OF A CAD MODEL SIMPLIFICATION FRAMEWORK FOR FINITE ELEMENT ANALYSIS

Brian Henry Russ, Master of Science, 2012

Thesis directed by: Professor Satyandra K. Gupta
Department of Mechanical Engineering,
Institute for Systems Research

Analyzing complex 3D models using finite element analysis software requires suppressing features/parts that are not likely to influence the analysis results, but may significantly improve the computational performance both in terms of mesh size and mesh quality. The suppression step often depends on the context and application. Currently, most analysts perform this step manually. This step can take a long time to perform on a complex model and can be tedious in nature. The goal of this thesis was to generate a simplification framework for both part and assembly CAD models for finite element analysis model preparation. At the part level, a rule-based approach for suppressing holes, rounds, and chamfers is presented. Then a tool for suppressing multiple specified part models at once is described at the assembly level. Upon discussion of the frameworks, the tools are demonstrated on several different models to show the complete approach and the computational performances.

The work presented in this thesis is expected to significantly reduce the man-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
2012

2. REPORT TYPE

3. DATES COVERED
00-00-2012 to 00-00-2012

4. TITLE AND SUBTITLE
Development of a Cad Model Simplification Framework for Finite Element Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland, College Park,College Park,MD,20742

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Analyzing complex 3D models using nite element analysis software requires suppressing features/parts that are not likely to in uence the analysis results, but may signi cantly improve the computational performance both in terms of mesh size and mesh quality. The suppression step often depends on the context and application. Currently, most analysts perform this step manually. This step can take a long time to perform on a complex model and can be tedious in nature. The goal of this thesis was to generate a simpli cation framework for both part and assembly CAD models for nite element analysis model preparation. At the part level, a rule-based approach for suppressing holes, rounds, and chamfers is presented. Then a tool for suppressing multiple speci ed part models at once is described at the assembly level. Upon discussion of the frameworks, the tools are demonstrated on several di erent models to show the complete approach and the computational performances. The work presented in this thesis is expected to signi cantly reduce the manual time consuming activities within the model simpli cation stage. This is accomplished through multiple feature/part suppression compared to the industry standard of suppressing one feature/part at a time. A simpli ed model speeds up the overall analysis, reducing the meshing time and calculation of the analysis values while maintaining and on occassion inproving the quality of the analysis.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**120**	

ual time consuming activities within the model simplification stage. This is accomplished through multiple feature/part suppression compared to the industry standard of suppressing one feature/part at a time. A simplified model speeds up the overall analysis, reducing the meshing time and calculation of the analysis values, while maintaining and on occassion inproving the quality of the analysis.

DEVELOPMENT OF A CAD MODEL SIMPLIFICATION FRAMEWORK FOR FINITE ELEMENT ANALYSIS

by

Brian Henry Russ

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2012

Advisory Committee:
Professor Satyandra K. Gupta, Chair/Advisor
Professor Hugh A. Bruck
Professor Santiago D. Solares

Acknowledgments

I would first like to express my deepest gratitude to my advisor, Dr. Gupta, who provided invaluable guidance during my tenure as his student. This guidance not only provided critical insight for the completion of the thesis, which otherwise would not have been possible, but also helped me develop as a person and an engineer. I would, also, like to thank Dr. Bruck and Dr. Solares for their support as committee members of my defense.

I am grateful for the support provided by Andrew Chorney. His expertise in model preparation and model analysis has proven to be most useful within my research, allowing me to generate practical solutions. I appreciate all the time he took out of his busy schedule to meet and answer any questions I had.

I owe a special thanks and acknowledgment to Madan Dabbeeru for his assistance and contribution to this work. Madan helped me extend my coding language and was always willing to help when a problem arose. Madan and my other lab mates also provided moral support and an entertaining work environment.

This research was funded by and done in collaboration with Patuxent River Naval Air Warfare Center and I would like to thank them for providing the financial support and the opportunities that arose from it.

Finally, I would like to thank my family, who have supported and believed in me from the beginning. Without you I would not have become the person I am nor even had the possibility to be given this opportunity.

Table of Contents

List of Tables v

List of Figures vi

List of Abbreviations viii

1 Introduction 1
1.1 Motivation . . . 1
1.2 Thesis Goals and Scope . . . 4
1.2.1 Part Simplification . . . 4
1.2.2 Assembly Simplification . . . 5
1.3 Organization . . . 6

2 Related Work 8
2.1 Overview . . . 8
2.2 Techniques Based on Surface Entities . . . 8
2.2.1 Boundary Loop Decomposition . . . 8
2.2.2 Low Pass Filtering . . . 10
2.2.3 Face Cluster Based Simplification . . . 10
2.3 Techniques Based on Volumetric Entities . . . 12
2.3.1 Voxel Based Simplification . . . 13
2.3.2 Effective Volume Based Simplification . . . 14
2.4 Techniques Based on Explicit Features . . . 17
2.4.1 Prismatic Feature Simplification . . . 18
2.4.2 Blend Simplification . . . 20
2.4.3 Arbitrary Shaped Feature Simplification . . . 22
2.5 Techniques based on Dimension Reduction . . . 23
2.5.1 Medial Axis Transform . . . 24
2.5.2 Mid-Surface Abstraction . . . 26
2.6 Sumamry . . . 27

3 Part Model Simplification for Finite Element Analysis 29
3.1 Overview . . . 29
3.2 Using Inductive Decision Tree to Generate Suppression Rules . . . 34
3.2.1 Training Parts . . . 38
3.2.2 Algorithm for Generating Rules . . . 41
3.3 Applying Learned-Rules for Simplification . . . 46
3.3.1 Extracting CAD Model Information . . . 46
3.3.2 Utilizing CAD Model Information within Rules . . . 48
3.3.3 Accounting for Parent-Child Relationships during Feature Suppression . . . 50
3.4 Results & Discussion . . . 54
3.5 Summary . . . 70

4 Assembly Model Simplification for Finite Element Analysis 72
4.1 Overview . . . 72
4.2 Part Suppression . . . 74
4.2.1 Part Utilization . . . 74
4.2.2 Material . . . 76
4.2.3 Algorithm . . . 78
4.3 Simplification . . . 82
4.4 Results and Discussion . . . 84
4.4.1 Modal Analysis . . . 84
4.4.2 Structural Analysis . . . 91
4.5 Summary . . . 98

5 Conclusion 99
5.1 Intellectual Contributions . . . 99
5.2 Anticipated Benefits . . . 100
5.3 Future Work . . . 101
5.3.1 Improvement to the Part Simplification Framework . . . 101
5.3.2 Advancements in Assembly Simplification Approach . . . 102

Bibliography 104

List of Tables

3.1 Training data collected from the examples shown in Fig. 3.4. 40
3.2 Rules for computing the suppressibility_state of holes 42
3.3 Rules for computing the suppressibility_state of holes 43
3.4 Rules for computing the suppressibility_state of chamfers 44
3.5 Rules for computing the suppressibility_state of rounds 45
3.6 *Errors and error types within the rule learning application: Conservative errors* can be allowed as they will not affect the analysis results but *Non-Conservative errors* can generate erroneous results. 49
3.7 Analysis of learned rules on the five test parts (A)-(E). 62
3.8 Analysis of models before and after simplification 63

4.1 Mesh improvement 86
4.2 Modes calculated within the modal analysis 86
4.3 Mesh improvement 93
4.4 Mesh improvement 97

List of Figures

1.1 Napier Sabre 24 cylinder aicraft engine [1] 2
1.2 *Suppressibility of a feature:* The decision to suppress or preserve a feature strongly depends upon the analysis context and the application. For a given loading and boundary conditions, "Insuppressible features" would affect the FEA analysis whereas the "suppressible features" would not. 5

2.1 Face clustering [18] 11
2.2 Original model construction [28] 17
2.3 Reconstruction of the model by rearranging the features and using the effective volumes [28] 17

3.1 Overview of the approach: Acquire knowledge through expert demonstrations, learn suppression rules from the acquired knowledge data, then apply the learned rules for feature suppression 30
3.2 *A Simple decision tree:* An example of a representative decision tree for suppressing a hole feature considering attributes diameter, minimum distance to the boundary condition feature, and the geometric type. Rectangular boxes and elliptical boxes represent the nonterminal nodes and the leaf nodes respectively. 35
3.3 Example of suppression based on inter-feature distances 38
3.4 Training examples used to collect attributes of holes, rounds, and chamfers. Synthetic parts are generated from these examples to tune the transition boundaries of the decision trees for better classification. 39
3.5 Suppression based on chamfer geometric types: (a) D × D (b) D1 × D2 (c) ANG × D (d) 45 × D (e) O × O and (f) O1 × O2 47
3.6 Dependency graph showing an example of finding features to be suppressed based on the parent-child relationships. Dependent relationships among the features should be considered before suppression; the suppression of one feature may suppress the dependent critical feature. 53
3.7 Part A: boundary condition features and feature suppression after rule implementation 56
3.8 Part B: boundary condition features and feature suppression after rule implementation 57
3.9 Part C: boundary condition features and feature suppression after rule implementation 58
3.10 Part D: boundary condition features and feature suppression after rule implementation 59
3.11 Part E: boundary condition features and feature suppression after rule implementation 60
3.12 Stress analysis results for the unsimplified and simplified models for part A 64

3.13 Stress analyses results for the unsimplified and simplified models for part B 65
3.14 Stress analyses results for the unsimplified and simplified models for part C 66
3.15 Stress analyses results for the unsimplified and simplified models for part D 67
3.16 Stress analyses results for the unsimplified and simplified models for part E 68
3.17 Example of suppression based on parent-child relationships 69

4.1 Overview of the assembly approach 72
4.2 Example assembly model 77
4.3 Stresses within the model when the two parts are made of the material steel 77
4.4 Stresses within the model when the two parts are made of the material nylon 78
4.5 Simple four-legged table for illustration of the parent-child relationship within a model 80
4.6 Example of the generated parent-child relationship graph 82
4.7 Unsimplified boom model for a modal analysis 84
4.8 Simplified boom model for a modal analysis 85
4.9 (a) Mode 1 of the unsimplified boom assembly (b) Mode 1 of the simplified boom assembly 87
4.10 (a) Mode 2 of the unsimplified boom assembly (b) Mode 2 of the simplified boom assembly 88
4.11 (a) Mode 3 of the unsimplified boom assembly (b) Mode 3 of the simplified boom assembly 89
4.12 (a) Mode 4 of the unsimplified boom assembly (b) Mode 4 of the simplified boom assembly 90
4.13 Unsimplified weight bench assembly model for a structural analysis . 91
4.14 Simplified weight bench assembly model for a structural analysis . . . 92
4.15 Von Mises Stress distribution on the unsimplified weight bench assembly 94
4.16 Von Mises Stress distribution on the simplified weight bench assembly 94
4.17 Unsimplified boom assembly model for a structural analysis 95
4.18 Simplified boom assembly model for a structural analysis 95
4.19 Von Mises Stress distribution on the unsimplified boom assembly . . 97
4.20 Von Mises Stress distribution on the unsimplified boom assembly . . 97

List of Abbreviations

CAD	Computer Aided Design
FEA	Finite Element Analysis
FE	Finite Element
ProE	Pro/Engineer
2D	Two Dimensional
3D	Three Dimensional
SR	Simplified Region
LPF	Low Pass Filtered
TDPS	Trihedral Discretized Polyhedra Simplification
MDCO	Maximal Division Classical Octree
BTG	Border Terminal Grey
LOD	Level of Detail
B-Rep	Boundary Representation
NMT	Non-Mainfold Topological
CSG	Constructive Solid Geometry
FCT	Feature Constructing Triangles
RAG	Region Adjacency Graph
MAT	Medial Axis Transform
API	Application Programming Interface
XML	Extensible Markup Language
AVMI	Air Vehicle Modification and Instrumentation
CG	Center of Gravity
FEA	Finite Element Analysis

Chapter 1

Introduction

1.1 Motivation

Engineering models today are mainly developed within a 3D computer aided design (CAD) software, for the reasons that the CAD models are unambiguous, visualization and manufacturing-planning is easier within the environment, and 3D model data exchange is well established. The 3D models created using these CAD systems often contain an abundant amount of features. Fig. 1.1 illustrates an aircraft engine with thousands of features. These features are necessary for a wide variety of life cycle needs including safety, functionality, aesthetics, and manufacturability.

The generated 3D CAD models are used for downstream applications within the design and manufacturing process. Finite element analysis (FEA), is one of the downstream applications, has emerged as an important engineering analysis tool and is extensively used for structural, thermal, and modal analyses to ensure that the design of the model will meet the specifications required from the operation of the model. FEA is a numerical technique for calculating approximate solutions of complicated equations used to calculate the analysis results. It does this by generating a mesh, grid of nodes, over the entire model. The density of the grip varies throughout the model, using having a higher density at fillets, corners, and

high stress areas. The mesh contains the structural and material properties, which define how the structure reacts to the conditions implemented within the analysis. At each node an equilibrium equation is generated and the equations are solved to obtain the results in the analysis.

Importing the detailed 3D models, such as the one shown in Fig. 1.1, into the FEA software often presents several challenges. First, the meshing time rapidly increases with the complexity of the model. Second, it may take a long time to analyze the large FEA mesh that results from the detailed CAD models. Third, the presence of a small feature may lead to a poor quality mesh and hence may lead to erroneous analysis results. In extreme cases, the automated meshing may fail on very complex models, requiring manual meshing of the model. Utilizing a more powerful computer can not solve all the issues associated with analyzing complex CAD models. Meshless methods have been proposed as an alternative to FEA, which use the geometry of the object directly for the calculations. These methods are not as widely used within the community.

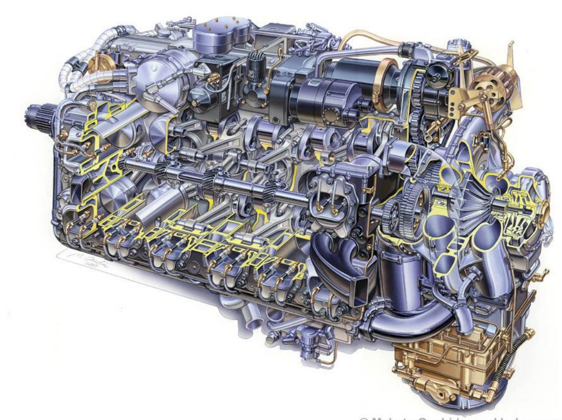


Figure 1.1: Napier Sabre 24 cylinder aicraft engine [1]

To circumvent the above described problems, experienced engineering analysts rarely perform meshing on the original complex models containing all the design features. In most situations, the model is simplified before proceeding with meshing. Simplification carefully examines the model and suppresses the features/parts that are not likely to influence the analysis results, but may significantly improve the computational performance both in terms of mesh size and mesh quality. Currently, the simplification of models is mainly performed manually, which can take a long time to perform on a complex model and can be tedious in nature. Also, most of the state-of-the-art mesh generation algorithms fail to generate meshes for those features that are smaller than the FE size. In these situations, CAD model simplification is essential to producing FE models adaptable to analysis. Certainly, automation in this task can improve analysts' productivity. Hence, automated feature suppression has been acknowledged as a useful tool in CAD-FEA integration [26].

Automated optimization of a CAD model for analysis can be a highly beneficial tool within the industry. As mentioned previously, the models are currently simplified manually for finite element analysis. Automating this process can potentially decrease the amount of time required for simplification, suppress all non-critical features/parts, and retain all critical fetaures/parts; ultimately speeding up the process and reducing the amount of errors involved within the process. For automation of this simplification process, numerous factors will have to be accounted for to correctly classify a feature/part as critical or non-critical to the analysis. A few of these factures include dimensions, material, and purpose.

Two different types of approaches can be taken to perform automated model

simplification for FEA. The first approach is geometry based and utilizes the geometric operators to simplify the parts [40]. Unfortunately, encoding the preferences of an analyst in this approach is very difficult and can lead to erroneous and inconsistent results [30]. The second approach is knowledge based. In this case, the preferences and knowledge used by the analyst are explicitly encoded as simplification rules. This approach captures analysts' knowledge in terms of rules that express the characteristics of features/parts that should be considered for suppression.

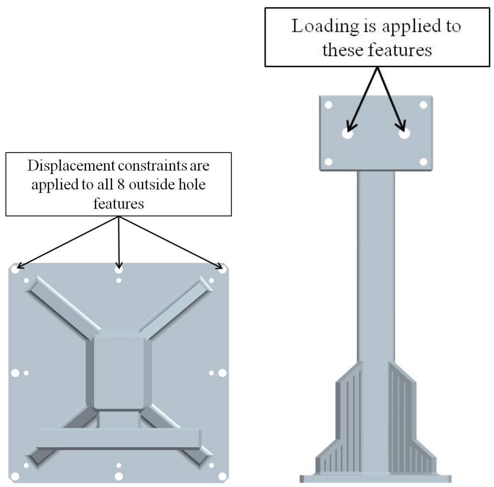
1.2 Thesis Goals and Scope

The goals and scope of this thesis revolve around developing a simplification tool for complex CAD models in preparation for FEA. The FEA types that are focused on within the framework are structural and modal. This tool is divided into two primary parts: part model simplification through feature suppression and assembly model simplification through part suppression and part simplification. For development of these tools a knowledge based approach has been utilized.

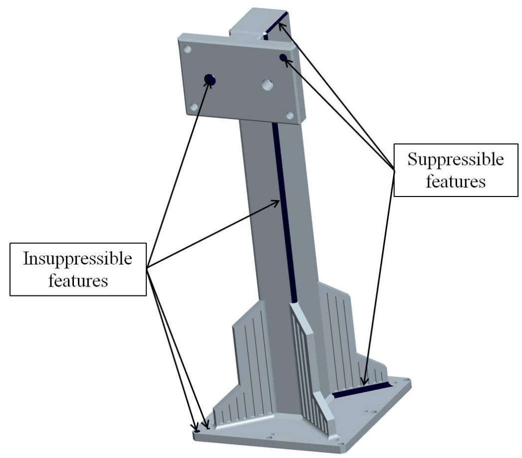
1.2.1 Part Simplification

Part model simplification consists of suppressing non-critical features within a CAD model. The determination of the non-critical (suppressible) features relies on the different attributes of not only the features themselves, but also of the entire part model and analysis. Some of these attributes include the feature type, feature dimensions, feature distances to boundary conditions, analysis type, and part di-

mensions. The dimensions of the part is required, so that the relative size of the feature to the overall part can be calculated. This is crucial, since part sizes are not limited and a feature that is seen as dimensionally small within one part may be quite large within another. Fig. 1.2 illustrates an example of a CAD model with features capable of being suppressed and those features that cannot be suppressed.



(a) Boundary conditions



(b) Suppressibility of features

Figure 1.2: *Suppressibility of a feature:* The decision to suppress or preserve a feature strongly depends upon the analysis context and the application. For a given loading and boundary conditions, "Insuppressible features" would affect the FEA analysis whereas the "suppressible features" would not.

1.2.2 Assembly Simplification

Assembly model simplification consists of suppressing or simplifying part models within a CAD model. Similarly to that of the part approach, parts can be labeled

as either non-critical (suppressible) or critical (non-suppressible); although unlike the part appproach, there can also be semi-critical parts within the assembly model. Semi-critical are the parts that must remain in the model for analysis, but the overall complexity of the part is not required; this can either consist of suppressing the non-critical features or simplifying the overall geometry of the part.

1.3 Organization

Chapter 2 is a survey of the current literature within the field of model simplification, which is not only limited to the simplification techniques specified for FEA model preparation. This chapter is arranegd into four collective sections; model simplification based upon surface operators, volume operators, dimension reduction operators, andexplicit feature operators. Each section is then further split up into the specific methods types of the simplification.

Chapter 3 presents a model simplification framework for CAD part models, by suppressing non-critical features. In this framework a decision-tree based representation has been developed for implementing the suppression rules on features; such as holes, rounds, and chamfers. Feature suppression demonstrations from an expert analysts provide a useful way of acquiring application dependent feature suppression rules. This method does not burden the human expert with explicitly cataloging feature suppression rules. Instead, feature suppression rules are extracted through demonstrations, in an unobtrusive fashion. These application dependent rules can be learned by using decision tree learning algorithms. We have developed

utilities in the Pro/Engineer CAD system to identify features that meet the learned rules and suppress them.

Chapter 4 presents a model simplification framework for CAD assembly models, by suppressing/simplifying non-critical parts. In this framework utilities have been developed within the Pro/Engineer CAD system to identify and suppress parts with a specified material or PART_NAME value.

Chapter 5 discusses the intellectual contributions of the research and possible areas of future research to further extend the frameworks discussed within the thesis.

Chapter 2

Related Work

2.1 Overview

Model simplification has been a focus within the modeling area for a while now, consequently there is a rich body of literature within this field. As not all of the literature can be reviewed within this thesis, I will focus on some of the more prevalent proposed techniques. For organization purposes the section follows the classification within a survey paper by Thakuer et al. [40], specifically: surface based operators, volume based operators, dimension reduction based operators, and explicit feature based operators.

2.2 Techniques Based on Surface Entities

This section reviews the techniques for which the simplification is based upon the surface entities and are organized into three classifications; boundary loop decomposition, low pass filtering, and face cluster based simplification.

2.2.1 Boundary Loop Decomposition

Sun et al. [39] presents a simplification method for B-rep models using region suppression. First, the simplified region (SR) and its boundary loop must be deter-

mined and classified. A hybrid method, both automated and manual, was adopted for determination of the SR and the faces which are to simplified are identified using their area ratio. After identifying the SR, the boundary loops are formed from the edges of the region. Each SR is classified according to whether it has single or multiple boundary loops. Multiple loop boundaries must be reduced to a single loop, either by dealing with the loops separately or constructing a bridge between the two separate loops to combine the loops into one.

A divide and conquer strategy is then used to decompose the loops. This recursively divides the boundary loops into multiple sub-loops by connecting two non-adjacent vertices with an edge, if there are either critical edges or boundary edges lying on the same underlying curve. The iteration process continues until the no co-defined edge pairs can be identified. The sub-loops are then further decomposed into co-surface loops.

Upon the decomposition of the boundary loops, the SRs are deleted from the model. The hole is fixed by REPAIR and PATCH operators. The REPAIR operator modifies the neighboring faces boundary and the PATCH operator constructs a new face and stitches the faces to the model. Both operations are possible through the use of a co-surface loop. To validate the B-rep model duplicate edges, vertices, and faces are merged. Advantages of the technique described is that it does not require Boolean operators and the REPAIR operator is insensitive to different types of features.

2.2.2 Low Pass Filtering

Lee and Lee et al. [29] proposed a semi-automatic low pass filtering technique using Discrete Fourier Transform. For implementation of the method the CAD model must first be digitized, converted into a 2D digital image of resolution 512 X 512. A surface can then be plotted from the pixels, having a value of -1.0, 0, or 1.0. Upon application of the Fourier transform to the surface function, the surface is expressed in the frequency domain. Within the frequency domain the low frequency terms represent the overall shape, whereas the high frequency terms represent the detailed features. Removal of the high frequency components generates a low pass filtered (LPF) model of the original model. The adverse result of the LPF model is that all sharp corners were rounded. Removal of the sharp edges is unfavorable in the analysis model, since sharp edges tend to acquire a higher stress concentration. To retain the sharp edges within the simplified model, the boundary elements average distance between the original model and LPF model is evaluated as a metric of complexity. If the metric is below a given threshold the entity is considered to be detailed and if the metric is below the given threshold the entity is considered to be to belong to the overall shape. The method is only developed for 2D images and if extended to a 3D image it is limited in terms reconstructing the removed faces.

2.2.3 Face Cluster Based Simplification

Inoue et al. [18] proposed a face clustering technique for CAD model simplification. The first step within the process is to decompose the model into faces. From

this decomposition a planar adjacency graph can be constructed; the graph nodes represent the faces, which are connected by arcs representing the adjacency of the face pairs. To determine the nodes to be clustered, a weight for each arc is calculated based on several geometric indices. The geometric indices attempt to take the area size, boundary smoothness, and region flatness into consideration. The equations for these indices are presented within the paper for reference.

Upon calculation of the arc weights, the arcs are prioritized for node clustering by ranking the weight values highest to lowest. After prioritizing the arcs, a mergeability test is completed using the geometric indices. The top weighted arc that passes the mergeability test is then contracted, merging the two connected nodes. If the two nodes both had an connected with a similar node, the two arcs are combined into one. This process is then completed for the entire adjacency graph, simplifying the original model. Fig. 2.1 display the process of removing the arcs and merging the nodes. For the example discussed within the paper, a model containing 1742 individual faces was simplified to a final model containing only 34 faces.

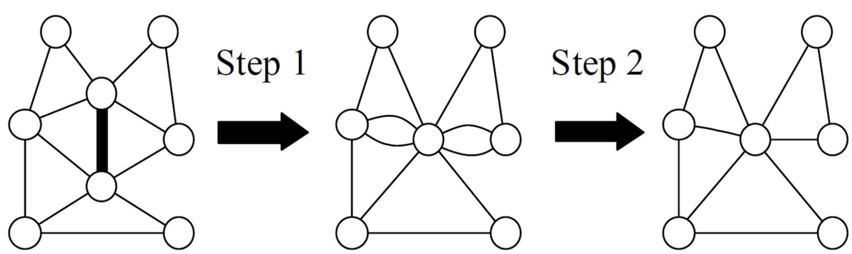


Figure 2.1: Face clustering [18]

Sheffer et al. [2] expanded upon this technique proposed by Inuoe. The clustering algorithm is maintained within the new process, although different geometric

indices are calculated. The new geometric indices quantify the boundary preservation, region size, region smoothness, and simplicity of the boundary shape. The main advantage of these indices is that they can also account for non-planar regions.

In addition to the new indices, Sheffer introduced a collapsibility test to determine faces that can be symetrically divided between adjacent clusters, after the clustering process. This division is completed using virtual topology operators to merge and collapse the faces. In this process the face cluster found to be divisible from the collapsibility test is then split into a number of faces equal to the number of adjacent neighboring faces. The new faces are then merged to the respective neighboring face from which an edge is shared. This further reduces the number faces within the model.

In a similar technique to the cluster division by virtual topology operators, Dey et al. [10] introduced a framework to eliminate the feature effects at mesh level. This technique identifies mesh entities with per quality metrics, based upon the dihedral angle measurement and aspect ratio of the mesh. After identification, the suppression of these entities is ensured to maintain geometric validity of the modified mesh and the entities are suppressed accordingly.

2.3 Techniques Based on Volumetric Entities

This section reviews the techniques for which the simplification is based upon the surface entities and are organized into two classifications; voxel based simplification and effective volume simplification.

2.3.1 Voxel Based Simplification

Andujar et al. [3] proposed a topology-reducing simplification algorithm called Trihedral Discretized Polyhedra Simplification (TDPS). This algorithm contains three steps: discretization, reconstruction, and face reduction. Maximal division classical octree (MDCO) is used to discretize the model. MDCO arranges eight octants, obtained by recursively subdividing the cubic universe into an 8-ary tree. The leaves of the tree falling completely in or out of the model volume are labeled as either black or white, respectively, and the leaves containing part of the boundary, partially in or out of the model, are labeled as grey. The grey leaves are then further subdivided until the parts are either black or white or the tree reaches a certain level.

Reconstruction step involves converting the MDCO octree representation into a 3D digital picture, set of points arranged in a regular grid, and extracting an isosurface from the digital picture. To compute the digital picture the MDCO is traversed to find the BTG nodes, boundary nodes. Each BTG node is subdivided into eight octants and the color at each octant corner is then labeled as either white or black, depending on the color of the node in contact with the corner. From this digital picture a surface, R, is extracted based upon an error bound in terms of the solid-Hausdorff distance, satisfying the condition that all white nodes are completely out of R and all black nodes are completely inside the R. The face reduction step further reduces the geometric complexity of R, using an edge collapse technique.

The authors report that TDPS has several advantages over other model simplification techniques; suitable for both parts and assemblies, it is not affected by input

surface degeneracies, and produces a two-manifold solids even with non-manifold inputs. Although there is a limitation of TDPS in that it is not intended for small error thresholds.

2.3.2 Effective Volume Based Simplification

Frameworks for multiresolution representations of solid models producing simplified models at various levels of detail (LODs) have been extensively researched. [26, 28, 6, 25] The focus of the research can be split into two separate approaches. The first approach uses conventional solid boundary representations (B-rep) [6] and the second approach uses non-manifold B-reps [28]. Conventional solid B-reps is advantageous since it can be implemented into commercial 3D CAD systems, due to the same data structures; although the computational cost of the boundary evaluation is very high, causing the method to be time consuming. Unlike the conventional B-reps, non-manifold B-reps are less time consuming and therefore will be explored further.

The non-manifold model is capable of representing any combination of wireframe, surface, solid, and cellular models in a unified data structure. It consists of three key modules: a feature-based modeling module, a feature-based idealization module, and a non-manifold topological (NMT) modeling kernel. The modules manage the database of the features, perform the detail removal and dimension reduction, and manipulate the geometric models, respectively. The detail removal within the idealization module consists of three steps to simplify the model. First all

idealization features with a design application domain are extracted from the master model. Then according to the given LOD criterion, most often the feature volume, the extracted idealization features are rearranged. If the criterion is based upon decreasing volume, features with a volume below the given threshold are considered for simplification. Finally, the LOD is selected manually to obtain the appropriate simplified model.[26]

Both the conventional and non-manifold B-rep approaches have a feature arrangement issue. Rearrangement of the features to generate the various LODs of the model will possibly result in a shape different from that of the original model shape, since the additive and subtractive Boolean operations are not commutative. Lee et al. [25] and Choi et al. [6] introduce a delta volume, redefined volume, if required, of the subtractive feature operations to ensure model similarity. The rearrangement algorithm applied within these approaches combines all additive features to generate the lowest resolution model and the delta volumes are subtracted from the model in descending order of volumes to generate the higher resolution models. This method is only applicable for the specified feature rearrangement, not for any arbitrary rearrangement.

Lee [28] expanded upon technique by introducing the concept of the effective volume of a feature, fractional or entire volume of the feature to provide the same result in spite of Boolean arrangement. This volume is calculated using Boolean algebra. Let V_i denote the volume of the solid primitive of a feature F_i, X_i denote a Boolean operation, and M_n denote the resulting model obtained by applying n Boolean operations between n+1 solid models:

$$M_n = \prod_{i=0}^{n} \bigotimes_i V_i, where \bigotimes_0 V_0 = \phi \bigotimes_0 V_0 \tag{2.1}$$

If the j^{th} Boolean operation $X_j V_j$ is moved to the mth position M_n can be represented as follows:

$$M_n = \left(\prod_{i=0, i \neq j}^{m} \bigotimes_i V_i \right) \left(\bigotimes_j \left(V_j - \sum_{l=1}^{m-j} \varphi \left(\bigotimes_j, \bigotimes_{j+1} \right) V_{j+1} \right) \right) \times \left(\prod_{i=m+1}^{n} \bigotimes_i V_i \right) \tag{2.2}$$

where,

$$\varphi(a, b) = \begin{cases} 0 & if a = b \\ 1 & if a \neq b \end{cases} \tag{2.3}$$

Figs. 2.2-2.3 illustrate the use of Eqns. 2.1-2.3 on a model generated by applying five form features [27]. The original model is shown in Fig. 2.2. If we alter the order of features from $F_0 \to F_1 \to F_2 \to F_3 \to F_4$ to $F_0 \to F_2 \to F_3 \to F_4 \to F_1$ by moving F_1 to the last position, the Boolean operation sequence changes from $V_0 - V_1 U V_2 - V_3 U V_4$ to $V_0 U V_2 - V_3 U V_4 - V_1$, where V_i is the respective volume of F_i. When just reorganizing the form features V_1 will just remove the volume for which it encompasses, i.e. removing V_2 and the connection of V_2 to V_0. Although, if we apply Eqns. 2.1-2.3 the effective volumes of F_0, F_1, F_2, F_3, and F_4 are calculated to be V_0, $V_1 - V_2 - V_4$, V_2, V_3, and V_4, respectively. Fig. 2.3 displays the rearranged feature model using the effective volumes of the features, which is identical the overall original model. This technique can allow for arbitrary rearrangement of the features and generate the same final shape as the original model.

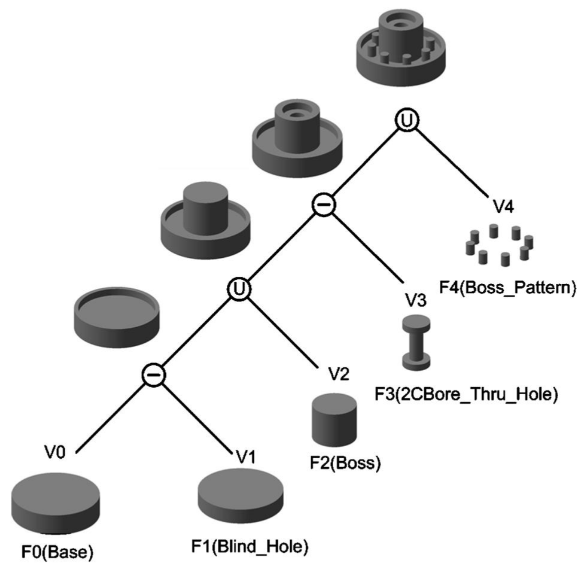


Figure 2.2: Original model construction [28]

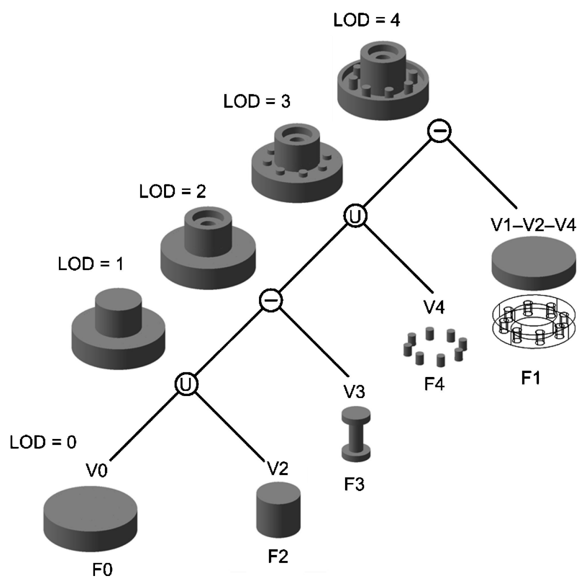


Figure 2.3: Reconstruction of the model by rearranging the features and using the effective volumes [28]

2.4 Techniques Based on Explicit Features

This section reviews the techniques for which the simplification is based upon the explicit features and are organized into three classifications; prismatic feature

simplification, blend simplification, arbitrary shaped feature simplification.

2.4.1 Prismatic Feature Simplification

Dabke et al. [8] implemented DESIDE-X, a rule-based software package for suppressing features, which relies upon a constructive solid geometry (CSG) tree and is specifically for modal analysis. Through the use of a feature by feature looping mechanism, each feature in the tree is analyzed and determined to be either significant or non-significant. The significance of the feature is evaluated based upon a rule base that takes into account different criteria; such as, mass and stiffness contributions, proximity to high stress areas, and the design functionality role. The nodes of the features found to be non-significant are removed from the tree along with the associated links. One drawback to this approach is that modern CAD systems rarely have the features readily available by means of a CSG tree.

Date and Nishigaki [9] present a alternative mesh simplification method through feature recognition and removal of blind holes, through holes, and bosses. The mesh is segmented into regions enclosed by the edges of the features. The edges are extracted using the dihedral angles between common edge faces. Upon mesh segmentation, the regions having an area larger than the given threshold are classified as base surfaces. The triangles not within the base surfaces are extracted as feature constructing triangles (FCTs). For each FCT the feature type and parameters are determined. the feature type is identified by the number of boundary loops and its concavity, while the dimensions are calculated using a least squares fit. Recognized

features are removed by deleting the FCTs and then filling the boundary loops by triangulation. An advantage to the proposed method is that the removed features are capable of being recovered later on in the process, which is extremely useful for a design model within the industry.

Gao et al. [14] proposed another framework for feature suppression based CAD mesh model simplification. There are three overall steps; segmentation of the mesh model, recognition of form features, and feature suppression.

Segmentation of the mesh model is achieved through an improved watershed method. The first step within the improved method is to refine the mesh by adding new vertices to the model, which allow for the model curvature to be calculated. Adding new vertices must not change the model geometry, so a different manner is used to add the vertices for all the triangle edges as feature edges and for triangle edges which are not all feature edges. Upon refining the mesh, the mesh is then segmented using a multi-descent strategy. The refined mesh of the segmented regions is then replaced with the original mesh of the model, so the original triangles represent the segmentation results. These segmentation results are over segmented due to the added vertices, therefore iterative region merging is used to generate the final segmentation.

After segmentation of the model is complete, a graph-based method recognizes the features within the model. Initially the region adjacency graph (RAG) is set up, where each node refers to a region and each arc represents the relationship between the regions. All base regions are then determined from the regions which have either an inner loop or concave vertex and deleted from the RAG. The remaining sub-

graphs of the RAG may then correspond to a form feature capable of suppression. To determine which ones are capable of suppression, the adjacency of the regions with the base features on inner loop edges or concave edges are checked and the sub-graphs that are adjacent is taken a s a candidate. Each feature graph candidate is compared to the RAG of the pre-defined features and a feature is recognized, if a match is found.

To generate the simplified model the features recognized in the previous step must then deleted and the hole filled. [14] proposed three different hole filling techniques planar Delaunay triangulation method for planar features, poisson equation method for curved features, and a blend feature method.

2.4.2 Blend Simplification

Joshi and Dutta [21] propose an algorithm to suppress blends with a circular cross-section. Recognition of the blend is determined from the curvature variation across the faces, calculated along the two directions, (U,V), of the parameterized NURBS surface, X(U,V).The curvatures at any point are given by Eqns. 2.4 and 2.5.

$$K_U = \frac{\left| \frac{\partial X}{\partial U} \times \frac{\partial^2 X}{\partial U^2} \right|}{\left| \frac{\partial X}{\partial U} \right|^2} \tag{2.4}$$

$$K_V = \frac{\left| \frac{\partial X}{\partial V} \times \frac{\partial^2 X}{\partial V^2} \right|}{\left| \frac{\partial X}{\partial V} \right|^2} \tag{2.5}$$

The face is marked as a potential blend, if K_u remains constant in the V direction and is greater than the calculated threshold and vice versa. The threshold is calculated based upon the object boundary box, causing the recognition to be independent of the model scale. For the potential blends to be classified as blends, the curves in the direction of the blend must be opposite an angle between 0 and 180 at the spine curve.

Blends are then suppressed in the reverse order of creation, to maintain a valid model. The blend order is determined by visiting each blend and then analyzing the adjoining faces to determine whether or not these faces are also blends. If the adjoining face is determined to be a blend, an evaluation is done to whether the two blends were created within the same operation or if the adjoining faces blend preceded/followed the initial blend operation. Suppression of the blend involves removing the blend face and then extrapolating the adjacent faces to a point of intersection, generating a simplified model.

Venkataraman And Sohoni [41] presented a similar algorithm for blend detection and order arrangement; although rather than considering the curvatures over the entire face along the parametric direction, [41] compares the curvatures of adjoining faces at the common edge midpoints. Then the suppression algorithm deletes the blend faces and collapses the cross edges, mutual edges of the blend face and adjacent edges, into vertices and the spring edges, the edges between the blend face and adjoin faces, into a single edge.

Zhu and Menq et al. [42] also present a method to suppress constant radius rounds and fillets, but within a B-rep model focusing on the repairing the suppressed

features. Initially, trace faces, faces generated from edge blending, are identified by the convexity of their topological entities. These trace faces are then removed from the model, suppressing the rounds and fillets. This suppression leaves gaps between the adjacent neighboring faces in the model. To close this gap a knitting process is implemented, which generates new edges and vertices. The overall knitting process is determined on the basis of on two factors of the H.E of the trace face, is the H.E. a disk or ring and is the H.E. a closed loop or open loop. This process has a couple advantages; the new B-rep model maintains geometric and topological consistency, along with being a reversible process.

2.4.3 Arbitrary Shaped Feature Simplification

Joshi and Dutta [21] proposed algorithms for the identification and suppression of different features in B-rep models. The first algorithm described is for ”hole” features, which are described as any loop of edges without a surface on the inside. Initially, the edges having a surface on one side and no surface on the other are labeled as free edges. Upon labeling the free edges, all the free edges are formed into closed loops; starting at one vertex and following the edge until the starting vertex is reached again. All these closed loops are considered to be holes, even the boundary of the model. The boundary must be eliminated as a hole, so to do this, the perimeters of all the closed loops are calculated and the hole with the largest perimeter value is designated as the model boundary; thus, removing the consideration for suppression. All loops with a perimeter value below the given

threshold are deleted and replaced by a surface patch with the same underlying surface equations as the loop.

Boss features, compound feature formed by a planar surface and two fillet rings, have also been implemented within the framework designed by [21]. For each planar face within the model, the adjacent faces on the outermost loop are visited to determine if they are a blend (defined within the blend section). If the adjacent faces are found to be blends and the blends are all part of the same loop, then the algorithm analyzes the adjacent faces of the opposing side of the blends. If the opposing faces are found to either have another blend loop on its other side or be a blend loop itself, then the faces are grouped together and labeled as a boss. Suppression of the boss entails removal of all the components and patching the surface, similar to the suppression of the holes.

These algorithms, in combination with the algorithm proposed for blends, can be extended to recognize and suppress any feature type in a model.

2.5 Techniques based on Dimension Reduction

This section reviews the techniques for which the simplification is based upon the dimension reduction and are organized into two classifications; medial axis transform and mid-surface abstraction. Dimensional reduction can be very beneficial for FEA. In some cases it is possible to reduce the dimension of the model without critically affecting the results of the model. This dramatically reduces the computational time of the model. For example, a long round slender bar of uniform diameter

can be reduced to a 1D beam [40]. Other methods have led to significant advances in thin structure model simplification, but may not be applicable to generic structures; such as an algebraic reduction for beams [19], nonlinear algebraic reduction for snap-fit simulation [20], and an algebraic reduction of 3D plates [31], a surface integration approach for 3D beams [36] and a method to decompose a model into parts and then apply the mid-surface abstraction technique by [7].

2.5.1 Medial Axis Transform

The medial axis transform, initially proposed by Blum [16], provides a skeleton-like representation of a geometric shape, which is comprised of a set of geometric entities. The geometric entities are generated by tracing out the centroid of the maximal inscribed disk/sphere as it rolls around the interior of the object. This concept provides both geometrical and topological proximity information, enabling a dimension reduction of the model, i.e. a 2D model can be reduced to a 1D model and a 3D model can be reduced to a 2D model.

Robinson et al. [35] discusses a medial axis transform method for both 2D and 3D models. Only the 3D reduction will be touched on within, but the paper can be referenced for the 2D or mixed dimension reduction. The first step in the process is to identify the thin regions within the model. In this step medial flaps, medial edge or face that terminates with a value of zero; medial faces not defined by two objects; and medial faces are with a large variation in 3D medial radius or thickness are all eliminated from the possibility of reduction. The remaining medial faces are

analyzed to determine if their lateral face dimensions are sufficiently large relative to their thickness. For this a conservative aspect ratio, since the smallest distance is divided by the largest thickness, is calculated from a 2D MAT on the 3D medial face. A medial face with an aspect ratio less than the critical aspect ratio is also eliminated from the possibility of dimension reduction. Not all medial faces that are left are capable of reduction, so only the capable regions should be reduced.

After determining the capable faces for reduction, mid-faces are created. Mid-faces are generated by an insetting of the interface by a distance equal to a variable OS multiplied by the region thickness, achieved by reusing the 2D MAT on each 3D medial face, and introducing rounding splines at convex corners of the inset. Upon generation of the mid-faces, the model is partitioned into thin sheet and complex regions. The bounding edges of the mid-face is project onto the medial faces and the projected area is deleted and replaced by a thin sheet. The thin sheet is then extended to the boundary of the component. Mixed dimensional coupling is required to at the interface of the detailed section and reduced section to join the elements.

The has been extensive research done for the MAT and only one of the processes has been discussed within this thesis. Other algorithms that have been presented are a method for the reduction of 2D planar or 3D shell models by a Donaghy et al. [11], polyhedral model homotopy preserving medial axis transform by Sud et al. [38], and an extension on the common MAT method known as Θ-MAT for polyhedral mesh models by Foskey et al. [13]. MAT has been extensively researched and the survey paper by Attali et al. [5] can be reviewed for a more complete review of MAT.

2.5.2 Mid-Surface Abstraction

Rezayat [34] introduced a mid-surface abstraction technique. Abstraction of the mid-surface involves four steps; pairing surfaces, surface topology and size based adjacency graphs, mid-surface patch generation by geometric interpolation and 2D Boolean operations, and adjacency based patch sewing. Within the surface pairing step, all the faces of the model are paired excluding the end-caps (faces on the edges of the solid model) and orphans (faces that are not on the thin-wall section). The excluded faces are determined by whether or not the ratio of the minimum face height and length divided by the thickness measure is greater than a given parameter X. A ratio less than X distinguishes a face as either an end face or orphan. To pair the faces an arbitrary face is selected as a seed face and a ray is cast into the material side in the normal direction of the face. The face upon which the ray hits is identified along with the common edge faces as processed. This is then repeated until all the faces have been processed.

Upon pairing the faces, an adjacency graph is then generated. The nodes of the graph are the faces and the connecting arcs are the relationship between the faces, either paired or stitched (common edge faces). The cyclic pattern of the adjacency graph references the geometric configuration of the model. A geometric interpolation of the paired faces is used to generate a mid-surface patch, where a 2x2 grid and a 15x15 grid are utilized to create the interpolation points of planar and free-form surfaces, respectively. The patches are sewn together at their intersecting edges, trimming the patch area beyond the intersection. If no intersection is possible, then

a new surface is created between the patches by a mesh of curves approximation. The part thickness, T, at a point, P, on the mid-surface of the parent surfaces, S1 and S2, is given by

$$T = dist\,(P, S2) + dist\,(P, S1) \tag{2.6}$$

The authors have reported several benefits of this method in comparison to the medial axis transform methods. These benefits include volume preserving, efficient feature suppression and detail removal, shape definition through geometric reasoning, and part form reflection.

2.6 Sumamry

The techniques described above are present many methods for which a model can be simplified for different applications. For the work discussed within this thesis, the techniques based on explicit features and dimension reduction are most useful; since our main focus is to suppress non-critical features for finite element analysis purposes. Within the category of explicit features we explain techniques for suppressing and modifying different features, such as: blind holes, through holes, chamfers, and rounds of any given geometric model. Although these methods discuss ways in which a model can be simplified, the literature does not provide error estimations between the analyses of the original and simplified models. For this [17] developed a formal theory for computing the defeaturing-induced engineering analysis errors, which would help to identify the suppressible features. [23] presents

an algorithm or computing the sensitivity of the design to finite feature suppression, thereby identifying the features that would affect the FEA. The approach discussed within this thesis has compared the analysis values, to ensure the technique is within an error range of 5% between the models.

Chapter 3

Part Model Simplification for Finite Element Analysis

3.1 Overview

The main focus of the approach is to design an automated tool that assists a user in identifying the features capable of suppression. The overall approach towards generating a CAD simplification tool, shown in Fig. 3.1, can be broken down into stages. First, the experts' knowledge is captured through demonstration and then suppression rules are generated with the use of a statistical induction learning technique [22]. In order to encode the experts' knowledge a representation is needed with the adequate expressivity. A representation was developed that allows the suppression rules to be expressed in a neutral format and these rules are expressed as a binary decision tree. The developed language has a very rich library, which allows the user to edit the rules if necessary. Upon learning the rules, they are applied for identification of suppressible features within the test parts. Features that are found to be suppressible are further examined based on their parent_child relationships. Lastly, the expert verifies the suppressibility of the features and the CAD model is simplified accordingly.

The decision to suppress or preserve a feature strongly depends upon the analysis context and the application. For developing a rule language the following four attributes that play a major role in determining whether or not the feature is

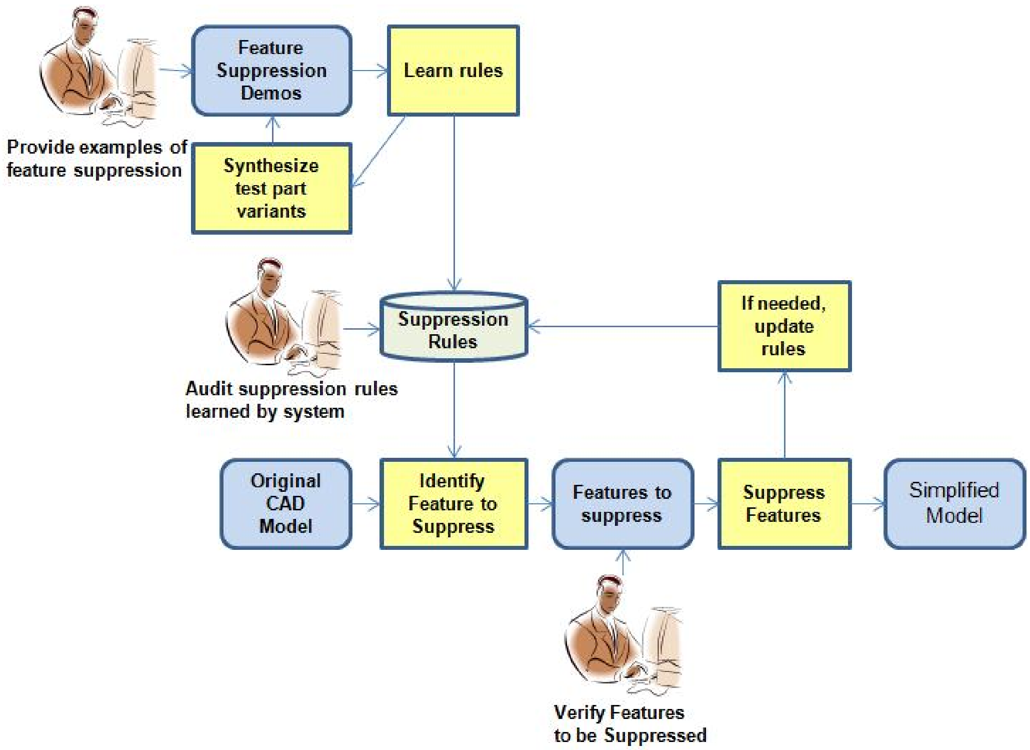


Figure 3.1: Overview of the approach: Acquire knowledge through expert demonstrations, learn suppression rules from the acquired knowledge data, then apply the learned rules for feature suppression

suppressed were considered:

1. Feature types play a major role in determining what features should be suppressed and what features should be preserved. Certain feature types are often more likely to be suppressed. For example, holes, fillets, and chamfers often do not significantly affect the analysis results. However, the presence of a single small hole in the model can significantly increase the size of mesh and also lower the quality of mesh element. So, such cosmetic features are prime candidates for suppression.

2. Feature dimensions play a major role in determining whether they should be suppressed or not. Often features with small sizes are prime candidates for suppression. However, in many cases it is not the absolute dimensions, but the relative dimensions of the features that determine whether they should be considered for suppression or not.

3. Another important consideration when selecting features for suppression is their topological proximity to the features on the part on which boundary and load conditions are applied. Features that lie close to either the point of load application or the displacement constraints should not be eliminated.

4. In CAD models, feature references can be used to define placement of other features. Such dependency among features may cause problems during feature suppression. For example, let us assume that a feature is eligible for suppression. However, if a large number of features use this feature as a reference, then suppressing this feature may require redefining features using other features as references. Not doing so may lead to deletion of those features as well.

Rules that describe which features are possible candidates for suppression can be quite complex in nature. As more complex parts are simplified, the rule library is expanded upon in the decision tree. Conditions in the decision tree can be expressed using the values returned by the following functions supported by our representation framework:

`Feature_Type` (*feature_id*): This function returns the type of the feature identified

by variable *feature_id*.

`Feature_Parameter_Value` (*feature_id, parameter_name*): This function returns the value of the feature parameter specified by variable *parameter_name*. For example, *parameter_name* can be set to `DIAMETER` to get the diameter of a hole feature.

`Find_Features` (*expression*): This function returns the list of features that meet the criteria specified by the variable expression. Expressions used as an input for this function can be composed by referring to features types, feature labels, and keywords. We also support all standard logical constructs. For example, expression `((Feature_Type(`*`this_feature`*`) == HOLE)` $\wedge$ `(`$\neg$ `(Label == BOUNDARY_CONDITION)` $\wedge$ `(DIAMETER` $<$ `10)))`returns all features in the model that are of type `HOLE`, do not have label as `BOUNDARY_CONDITION`, and have `DIAMETER` values smaller then 10. Values used in the expression are assumed to have the same units as the units being used in the CAD model.

`Feature_Bounding_Box` (*feature_id*): This function returns the bounding box of the feature.

`Model_Bounding_Box()`:This function returns the bounding box of the CAD model.

`Max_Distance_Between_A_Feature_And_Feature_List` (*feature_id, feature_list*): This function returns the maximum distance of the feature defined by variable feature_id with the features in the *`feature_list`*.

`Min_Distance_Between_A_Feature_And_Feature_List` (*feature_id, feature_list*): This function returns the minimum distance of the feature defined by variable *feature_id* with the features in the *feature_list*.

`Parent_Features` (*feature_id*): This function returns the list of features that are used as a reference for constructing the feature defined by variable *feature_id*.

`Child_Features` (*feature_id*): This function returns the list of features that use the feature defined by variable *feature_id* as a reference.

The variable *this_feature* refers to the feature being currently evaluated. The rule language was generated based upon the four attributes discussed earlier; feature type, feature dimensions, topological distance, and feature references. The feature dimensions and topological distance are values that are relative to the overall size of the 3D model. To acquire a representative rule base for 3D models, these values must be normalized to eliminate the correlation between the overall model size and the feature values. To accomplish this task the feature values were divided by the average of the sides of the overall models bounding box. This normalizes the data and removes the dimensional bias of the of the feature value with respect to the overall model size.

The representation allows a wide variety of knowledge associated with feature suppression in a wide variety of contexts to be easily captured. A decision tree in the context of brackets and mounts to illustrate how the approach works has been developed. Section 3.2 describes the framework to learn the rules for suppression based on the knowledge/data collected during the expert demonstrations. Section 3.3 presents an algorithm for applying these learned rules and Section 3.4 presents test results of the simplification process.

3.2 Using Inductive Decision Tree to Generate Suppression Rules

Machine learning mechanizes the acquisition of knowledge from experience through the use of computational methods [24]. It aims to improve performance by automating manual time-consuming activity within the knowledge process and increase accuracy by discovering and exploiting regularities in training data [24]. A popular technique in robotics is learning from demonstration [4]. Inductive decision trees have proven to be a successful tool within the numerous fields and these are proven to be advantageous in improving the accuracy or efficiency of the system. Many real world problems have implemented the technique and their experiments have shown that the learned rules are more accurate than the handcrafted ones previously used [15, 12].

The framework implemented here requires the rules to classify the features as suppressible or non-suppressible. The feature attributes used for the rule learning process are both discrete, having a predefined set of options, or continuous, real numbers. Regression and other model-fitting methods require discrete attributes to be encoded as numbers, in which the transformation introduce spatial connotations that may affect the outcome. On the other hand Bayesian methods require all attributes to be discrete, so continuous attributes must be divided into sub-ranges [32]. For these reasons decision trees were used, since it can cope with both discrete and continuous attributes. Trees are also very clear and concise in style; along as context sensitive, allowing for the relevance of different attributes to be conditional on the outcomes of previous tests [32].

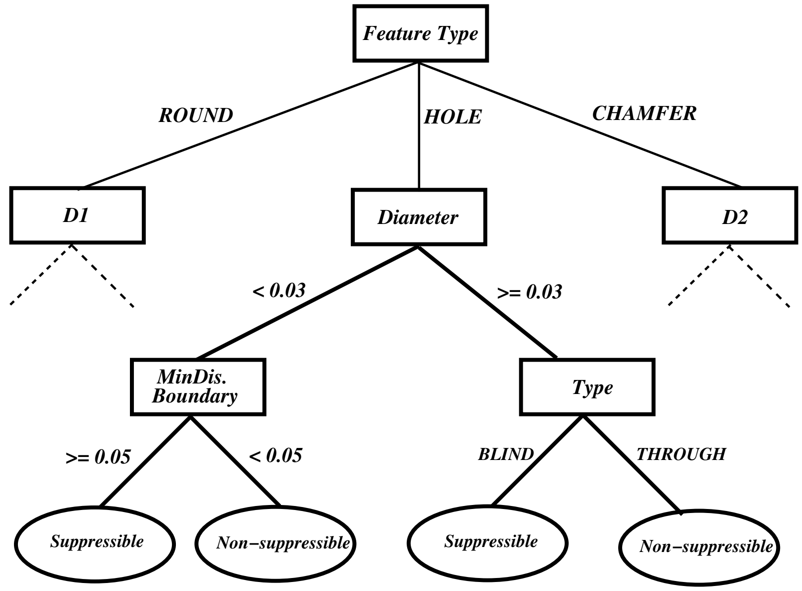


Figure 3.2: *A Simple decision tree:* An example of a representative decision tree for suppressing a hole feature considering attributes diameter, minimum distance to the boundary condition feature, and the geometric type. Rectangular boxes and elliptical boxes represent the non-terminal nodes and the leaf nodes respectively.

The decision tree is used to classify the suppressibility of the features in the following manner. Given a feature to classify, the decision tree is traversed from the root node to a leaf node along a path. All non-terminal nodes in the decision tree only define test conditions. These nodes in the decision tree point to exactly two children nodes. The left node is the node that should be executed if the condition is true. The right node is the node that should be executed if the condition is false. Terminal nodes, leaf node, in the decision tree assign the variable suppressibility_state for the feature. If the feature is suppressible, then the variable suppressibility_state is set to TRUE. Otherwise, this label is set to FALSE. For a

given set of rules as shown below, a simple decision tree can be formed (Fig. 3.2) by using various split conditions. The decision to suppress or not to suppress can be made based on the attribute values. Features that are found to be suitable for suppression are further examined based on their parent-child relationship.

- `If((Feature_Type(`*this_feature*`)==HOLE)` $\wedge$ `(Feature_Parameter_Value` `(`*this_feature*`,DIAMETER)` $<$ `0.03)` $\wedge$ `((Min_Distnace_Between_A_Feature_And_Feature_List` `(`*this_feature*`,Find_Features (Label==BOUNDARY_CONDITION)))` $>=$ `0.05) then (Suppressibility_State = TRUE))`

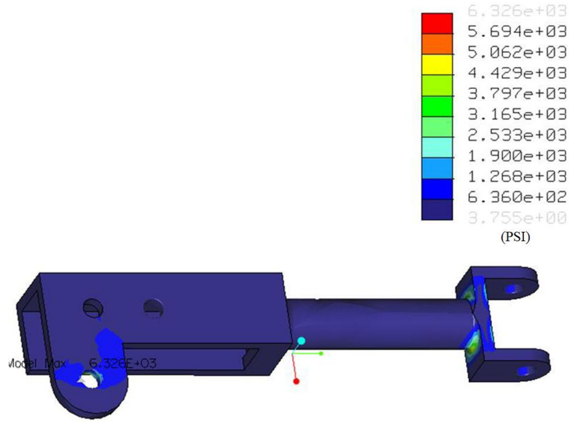
- `If((Feature_Type(`*this_feature*`)==HOLE)` $\wedge$ `(Feature_Parameter_Value` `(`*this_feature*`,DIAMETER)`$>=$`0.03)` $\wedge$ `(Feature_Parameter_Value` `(`*this_feature*`,TYPE)`$\neg$ `BLIND)` $\wedge$`(Feature_Parameter_Value` `(`*this_feature*`,TYPE)`$\neg$ `THROUGH) then` `(Suppressibility_State = TRUE))`

The set of attributes used for the learning process will be dependent upon the feature type. Holes have the feature parameters of diameter, type (through/blind), and minimum distance to the boundary conditions. The diameter and minimum distance to the boundary condition attributes are both continuous in nature, whereas the type attribute is discrete in nature. Rounds have the attribute types of radius, conic value and minimum distance to the boundary conditions. Chamfers have the attribute types of D1, D2, and minimum distance to the boundary conditions. Even though the attributes for rounds and chamfers are all considered to be continuous

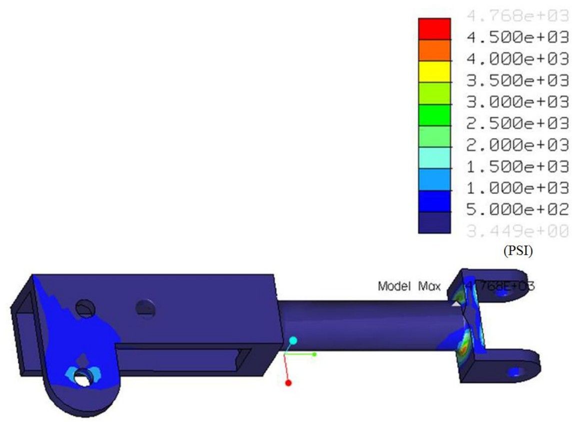
in nature, a decision tree is still used for analysis consistency with the hole features.

These attributes were selected to convey the key aspects of the features; feature type, feature size, and minimum geometric distance to the boundary conditions. These aspects have a effect on whether the feature is critical or not. A couple models have been analyzed to demonstrate the effects of the different aspects on the analysis results. The importance of knowing the minimum distance to the boundary conditions is illustrated in Figs.3.3(a)and(b). There is a round, meeting the dimensional requirements for suppression, but it is within a critical distance to the boundary condition. Suppression of this round feature is detrimental to the accuracy of the results. Upon suppressing this round feature the critical stress value changes from a value of 42.27 MPa within the original model illustrated in 3.3(a) to a value of 32.87 MPa shown in 3.3(b). When the round is maintained the critical stress value is comparable to the original model at a value of 43.61 MPa. Not only is value of the critical stress erroneous, but the location of the critical stress value has changed and can also be seen in the following figures.

To learn the decision trees the analysts experiential knowledge was obtained by conducting numerous simplification demonstrations on various training parts, where the information about the features and their associated suppressibility states assigned by the analyst was collected. Collecting the data of the non-suppressed features along with the suppressible features allows for a more expansive rule base for easy classification.



(a) Original model



(b) Simplified model with critical round suppressed

Figure 3.3: Example of suppression based on inter-feature distances

3.2.1 Training Parts

Formation of reasonably good decision trees can be possible by generating a large number of training parts for demonstration purposes; although this can be manually time consuming for the expert to conduct demonstrations for such a large data set. However, with the most informative training examples the decision tree can classify the features correctly without a need of as many examples, through

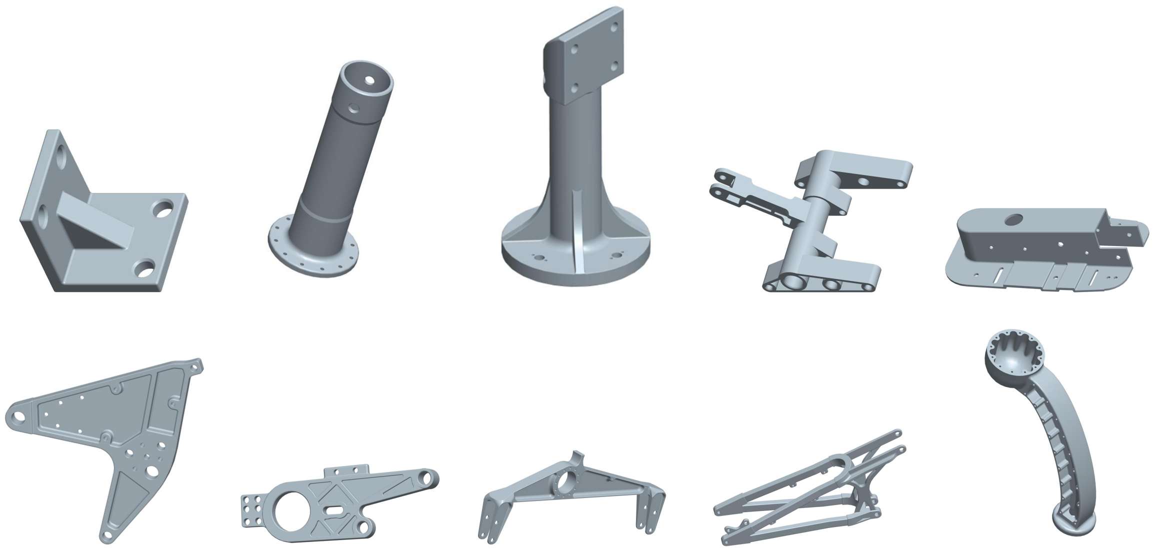


Figure 3.4: Training examples used to collect attributes of holes, rounds, and chamfers. Synthetic parts are generated from these examples to tune the transition boundaries of the decision trees for better classification.

the use of synthetic parts to refine the transition point values at the classification boundaries. For this, some heuristics were used for choosing places where the decision tree performs poorly and generate an additional set of training data at those places for refining the boundaries.

Initially, 10 training parts were started with that offer diversity in terms of feature types and feature parameters. A decision tree is formed from these initial parts. This decision tree failed to give the correct answer for those places where the transition region between the suppressible and non-suppressible class is very large and hence locating the correct transition boundary separating these two classes is a difficult task. To overcome this issue the divide-and-conquer strategy (binary search)

Feature Type	**No. of of Features**	**No. of Suppressible Seatures**	**No. of Non-Suppressible Features**
Holes	169	60	109
Chamfers	57	23	34
Rounds	118	75	43

Table 3.1: Training data collected from the examples shown in Fig. 3.4.

was used to locate this region and then tune the decision tree with the synthetic test cases generated in the vicinity of the boundary. To determine the transition region another 37 synthetic parts were generated to refine the transition points of the rule base. From these 37 synthetic parts the only recorded features were those features used for the transition region or features that required alteration when designing the transition feature of the synthetic part. Only recording these features within the synthetic parts eliminated the chance of biasing the data, through the introduction of the same features numerous times within the data. Since the data is being recorded in this fashion, there will be a relatively low total feature count for the number of total training parts. The training parts can be seen in Figs. 3.4. Numerous hole, round, and chamfer features were obtained from these examples for rule learning. There was a total of 344 features within the training parts, 158 were suppressible and 186 were non-suppressible. Table 3.1 displays the statistics of these features within the training parts.

3.2.2 Algorithm for Generating Rules

In this work, a decision trees was built from the geometric feature attributes and the associated suppressibility values. From the set of training parts a decision tree [22] can be formed to generate a classification model that can generalize beyond the training samples so that new samples can also be classified with a high accuracy. In order to do this, the decision tree must capture some meaningful relationships between the output class ($C = C_1, C_2, ..., C_k$) and its input data space consists of values from independent parameters ($A = A_1, A_2, ...A_k$). Within the data there are three independent parameters; for instance a hole feature utilizes the diameter, type, and minimum distance to a boundary condition feature. The dependent parameter, C_i, represents the class, whether suppressible or non-suppressible.

Development of the tree from the training cases begins at a test node, known as the root node, of the tree. The tree builds from the root node, splitting the dataset of one particular attribute at each sub-node. The division of the attributes is determined based on the information gain, which is maximized at each node; the attribute with the highest information gain is placed at the node. The information gain is the entropy relationship of the node before and after the split based on the specified attribute [22]. This process is continued until the node contains only one class; then the node is no longer a test node, but considered to be a leaf node. The tree is now considered to be fully grown. Fully grown trees tend to over-fit the data, meaning that the tree characterizes too much detail or noise from within the training set. To reduce the error of over-fitting the data, the tree was pruned. Pruning the

tree involves calculating the error rates of the branches of the tree. Comparison of the error rates determines whether the branch is left unchanged, changed to a leaf node, or replaced by an alternate sub-tree.

The generated decision trees for the different feature types are then written in the framework rule language. Tables 3.2-3.5 display the actual rules produced from the decision trees of the hole, round, and chamfer features.

Rule	**Suppressibility_State**
• If((Feature_Type(*this_feature*) == HOLE) ∧ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features (Label == BOUNDARY_CONDITION))) <= 0.02368)	FALSE
• If((Feature_Type(*this_feature*) == HOLE) ∧ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) > 0.02368) ∧ (Feature_Parameter_Value (*this_feature*,DIAMETER) <= 0.08451) ∧ (Feature_Parameter_Value (*this_feature*,TYPE) == THROUGH)	TRUE
• If((Feature_Type(*this_feature*) == HOLE) ∧ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features (Label == BOUNDARY_CONDITION))) > 0.02368) ∧ (Feature_Parameter_Value(*this_feature*,DIAMETER) <= 0.08451) ∧ (Feature_Parameter_Value(*this_feature*,TYPE) == BLIND) ∧ (Feature_Parameter_Value (*this_feature*,DIAMETER) <= 0.018626)	TRUE

Table 3.2: Rules for computing the suppressibility_state of holes

Rule	Suppressibility_State
• If((Feature_Type(*this_feature*)== HOLE) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label==BOUNDARY_CONDITION))) $>$ 0.02368) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $<=$ 0.08451) $\wedge$ (Feature_Parameter_Value (*this_feature*,TYPE)==BLIND) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $>$ 0.018626) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $<=$ 0.02314)	FALSE
• If((Feature_Type(*this_feature*) == HOLE) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $>$ 0.02368) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $<=$ 0.08451) $\wedge$ (Feature_Parameter_Value (*this_feature*,TYPE) == BLIND) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $>$ 0.018626) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $>$ 0.02314)	TRUE
• If((Feature_Type(*this_feature*) == HOLE) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $>$ 0.02368) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $>$ 0.08451) $\wedge$ (Feature_Parameter_Value (*this_feature*,TYPE) == THROUGH)	FALSE
• If((Feature_Type(*this_feature*) == HOLE) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $>$ 0.02368) $\wedge$ (Feature_Parameter_Value (*this_feature*,DIAMETER) $>$ 0.08451) $\wedge$ (Feature_Parameter_Value (*this_feature*,TYPE) == BLIND)	TRUE

Table 3.3: Rules for computing the suppressibility_state of holes

Rule	Suppressibility_State
• If((Feature_Type(*this_feature*) == CHAMFER) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $<=$ 0.004121)	FALSE
• If((Feature_Type(*this_feature*) == CHAMFER) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $>$ 0.004121) $\wedge$ (Feature_Parameter_Value (*this_feature*,D1) $<=$ 0.01714)	TRUE
• If((Feature_Type(*this_feature*) == CHAMFER) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label == BOUNDARY_CONDITION))) $>$ 0.004121) $\wedge$ (Feature_Parameter_Value (*this_feature*,D1) $>$ 0.01714)	FALSE

Table 3.4: Rules for computing the suppressibility_state of chamfers

Rule	Suppressibility_State
• If((Feature_Type(*this_feature*) == ROUND) $\wedge$ (Feature_Parameter_Value (*this_feature*,RADIUS) $<=$ 0.02623) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label==BOUNDARY_CONDITION))) $<=$ 0.013545)	FALSE
• If((Feature_Type(*this_feature*) == ROUND) $\wedge$ (Feature_Parameter_Value (*this_feature*,RADIUS) $<=$ 0.02623) $\wedge$ (Min_Distnace_Between_A_Feature_And_Feature_List (*this_feature*,Find_Features(Label==BOUNDARY_CONDITION))) $>$ 0.013545)	TRUE
• ((Feature_Type(*this_feature*) == ROUND) $\wedge$ (Feature_Parameter_Value (*this_feature*,RADIUS) $>$ 0.02623))	FALSE

Table 3.5: Rules for computing the suppressibility_state of rounds

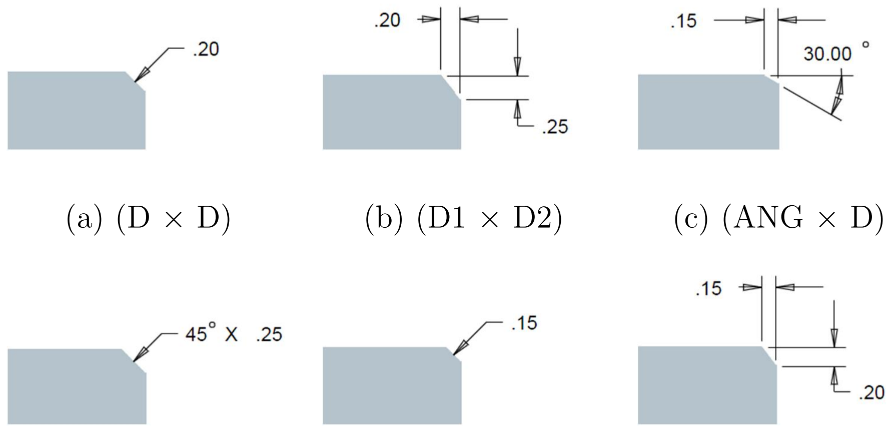
3.3 Applying Learned-Rules for Simplification

The rules have been learned from training parts, now the next step within the approach, as seen in Fig. 3.1, is to apply the rules to new models. To do this, a utility was needed to extract the attribute information and classify the features within the model, for this ProE was used.

3.3.1 Extracting CAD Model Information

ProE has a Pro/Toolkit API, which allows customization within the framework. In this API, the design model is stored by ProMdl, a data handle to retrieve the complete information of the model such as features, their dependencies, etc. By utilizing the ProMdl the framework traverses through all the features, where each feature can contain a large amount of information. In ProE, the complete design model is stored as a feature element tree that contains all information about the feature, including, for example: all attributes, such as the material, placements of holes; all references to the existing geometries such as placement references; and all dimension values. For example, a "hole" feature contains information as to whether the hole is straight, standard, or sketched; whether the hole is a clearance hole or a threaded hole; parameters of the hole, etc; all this information can be obtained through the element tree.

To implement the part of the rule Feature_Type(*feature_id*), the element tree of the complete design model was traversed and determined the feature f corresponding to the feature_id. Now, the element tree of this feature f was traversed to extract



(a) (D × D) (b) (D1 × D2) (c) (ANG × D)

(d) (45 × D) (e) (O × O) (f) (O1 × O2)

Figure 3.5: Suppression based on chamfer geometric types: (a) D × D (b) D1 × D2 (c) ANG × D (d) 45 × D (e) O × O and (f) O1 × O2

the internal structure of the feature information, such as the diameter value for a hole or the D1 value for a chamfer. There are many types of features that can be present within a CAD model, including holes, rounds, chamfers, fillets, steps, and slots. Currently, the framework implemented considers (a) chamfers (b) holes and (c) rounds. In the case of chamfers there is an option of six different geometric representations; described by one distance (D × D) (Fig. 3.5)(a); by two distances (D1 × D2) (Fig. 3.5)(b); by an angle and a distance (ANG × D) (Fig. 3.5)(c); by a 45^o angle and a distance (45 × D) (Figure 3.5)(d); by an off-set distance (O × O) (Fig. 3.5)(e); or by two off-set distances (O1 × O2) (Fig. 3.5)(f). In a similar manner the suppressibility_state of holes is described by two types; blind holes and through holes; and rounds are described as either conical or circular.

In addition to the size and type, feature locations must also be calculated

for the rule implementation. Feature locations play an important role in determining the extent of the feature criticality, since often features that are very close to the boundary conditions are very critical and should not be eliminated irrespective of their size. In most situations, closeness in terms of the topological distance is more important than the geometric distances. Features that are far away from the boundary conditions can also become critical if they are located within the stress flow path from the load application to the displacement constraints. Utilizing the available ProE API functions each and every feature of the design model was visited, computing the inter-feature distances among all the relevant feature pairs available in the model. Then the feature pairs are retrieved associated with load or displacement constraints locations. In the framework, features associated with load or displacement constraint locations are labeled as `BOUNDARY_CONDITION` features. This labeling needs to be done manually.

3.3.2 Utilizing CAD Model Information within Rules

The collected attribute data for each feature in the model from section 3.3.1 was normalized just like the training data, to remove the dimensional bias of the parts. The normalized data was then evaluated utilizing the machine learned rules, generated from the training data. For instance, the attribute values of a hole, consisting of the diameter, type, and minimum distance to the boundary condition, collected in section 3.3.1 were compared to the rule values in Table 3.2. Referring to Table 3.2 a hole with a calculated normalized minimum distance to the boundary

condition features of 0 would be classified as non-suppressible without any other attribute analysis, where as a normalized distance of .03 would then cause the diameter attribute to be analyzed. Upon analyzing the diameter, the type would then be used for comparison and depending on the diameter and type values the analysis could potentially stop there or continue. After the feature's attribute values were analyzed using the generated rule set, the features were then labeled as suppressible or non-suppressible determined by the classification obtained from the rule base.

Actual Suppressibility_State	**Rule Suppressibility_State**	**Classification**	**Error Type**
True	True	Correct	None
False	False	Correct	None
True	False	Incorrect	Conservative
False	True	Incorrect	Non-Conservative

Table 3.6: *Errors and error types within the rule learning application: Conservative errors* can be allowed as they will not affect the analysis results but *Non-Conservative errors* can generate erroneous results.

Implementation of the learned rules from the training data may sometimes be flawed, which can generate two different error types, shown in Table 3.6 when compared with the actual suppressibility_states of the features. The first type is a *conservative error*, which is an error that will not affect the analysis results of the simplified model. This occurs when the rule states not to suppress a feature that has an actual suppressibility_state of TRUE, or non-critical feature. The other type is a

non-conservative error, which will likely affect the analysis results of the simplified model. This error occurs when the rule states to suppress a feature that has an actual suppressibility_status of FALSE, or critical feature. For the reason that non-conservative errors can generate erroneous results, the error has been limited within the inductive learning process to allow for only conservatives errors. This was done through the use of the synthetic parts to refine the transition points.

3.3.3 Accounting for Parent-Child Relationships during Feature Suppression

Design model generation using a feature-based approach allows flexibility to compile a set of information about the product at higher levels of abstractions. This information can be represented in the form of data structures that maintain the associations between the geometric entities, further aiding in making any changes to the design. All major parametric CAD modeling systems have adopted the feature-based design approach by representing the final design model with features dependent on one another. This is called the design feature history tree [33, 37]. The same design model may be designed in several different ways, depending on the designer, who builds the design feature history. For the finite element analysis the order in which one simplifies the features is immaterial. However, it is necessary to understand the dependent relationships among the features, as the suppression of one feature may suppress another feature that may be very critical to the analysis. Hence, it is imperative to understand features' parent-child relationships for suppressions. For

example, on a block a hole was first created and then a round operation was performed on it. The round is defined as the child of the parent hole, which in turn is the child of the parent block. In this case, suppressing the hole would also suppress the round. Extracting these parent-child dependencies is crucial in order to avoid suppressing any non-critical features.

In the approach, the feature dependencies were initially extracted from the model using the available API functions and represented as a feature dependency graph, as illustrated in Figure 3.6. Let $G(V, E)$ represent the parent-child relationship graph. Each v in V represents a feature in the feature tree. Each element e in E is a directed edge (v, v) that represents the fact that node v is a parent of node v. Since features are inserted one by one in the feature tree and can only use the already defined features as references, by the property of the feature tree construction, G is a directed acyclic graph.

Within the graph the nodes represent the various features of the model. Each dependency is represented by an arrow, which is directed from the parent feature to the child feature. The tree illustrates the dependencies for easy analysis. After obtaining the G value, we follow an algorithm for identifying different cases of suppression possibilities, as a feature should not be suppressed just because it is suppressible. It may have child features indicating that it is not suppressible. On the other hand, it can be suppressed if all of its direct children can be suppressed. Hence, we needed to use a special sequence to determine which features can be suppressed by accounting for their parent-child relationships.

The algorithm for suppressing features is as follow:

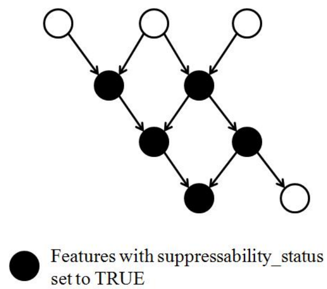
1. Construct G by inserting a node in V corresponding to every feature in the feature tree. If feature f' uses a feature f as a reference, then insert an edge (v, f') in G. Here vertex v corresponds to feature f and vertex v' corresponds to feature f'.

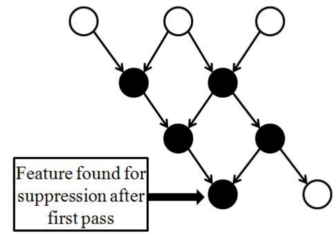
2. Find nodes X in V that do not have parents and have label suppressibility_state = TRUE. If X is empty, then reset suppressibility_state for all nodes to be FALSE and stop. Otherwise do the following:

   (a) Remove every node in X from G by deleting associated vertex and edges.

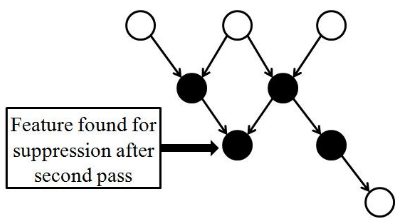
   (b) Go to Step 2.

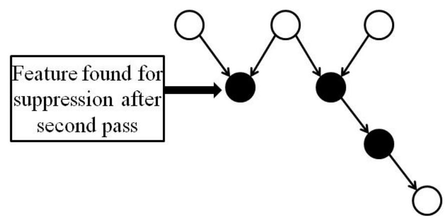
Figure 3.6(a) shows a parent-child relationship graph. Figures 3.6(b) through (e) show how executing the above algorithm correctly identifies the features to be suppressed. This algorithm discovers at least one feature to suppress; otherwise it terminates in Step 2. Hence, it takes at most V iterations to complete its task. In this way, what our tool is capable of suppressing the features based on the feasibility dependencies.



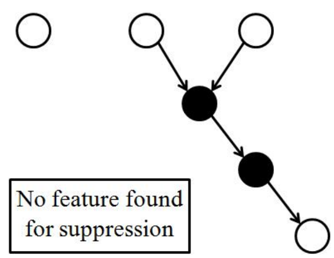


(a) Initial tree (b) First pass for suppression





(c) Second pass for suppression (d) Third pass for suppression



(e) Fourth pass for suppression

Figure 3.6: Dependency graph showing an example of finding features to be suppressed based on the parent-child relationships. Dependent relationships among the features should be considered before suppression; the suppression of one feature may suppress the dependent critical feature.

3.4 Results & Discussion

To demonstrate the simplification tool five parts were considered Fig.3.7-Fig.3.11. The simplified models were obtained by using the learned rules from the training models. Now a comparison was done of the analysis results of the original and simplified models (Fig. 3.7(b) - Fig.3.11(b). In terms of analysis we initially add displacement constraints and applied forces were added to each part and a finite element analysis was conducted with the help of Pro/Mechanica (available in ProE). Pro/Mechanica requires every model to contain a constraint and a force for analysis.

The location of the displacement constraints along with the applied forces is explained below. Figs. 3.7(a) - Fig.3.11(a) illustrate the boundary conditions (constraints and forces) applied to the to the five complex models.

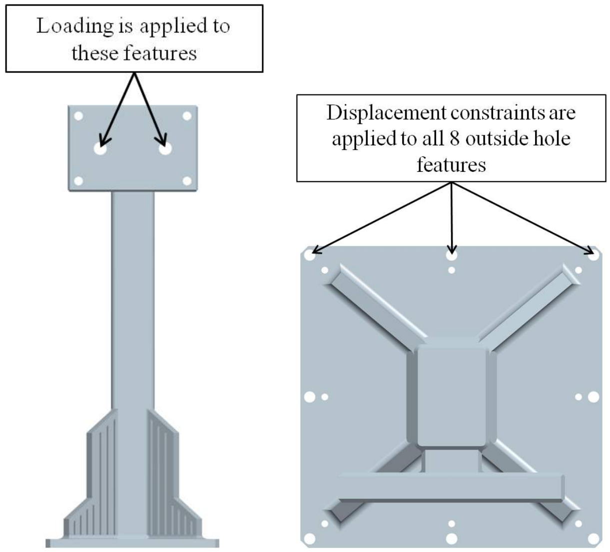
1. **Part-A**: The constraints were placed at the eight outside holes on the base of the part, where it is bolted into place. The forces were applied in the downward vertical direction located at the two center holes of the extension at the top of the part (Fig. 3.7(a)).

2. **Part-B**: The constraints are located at the vertical slots. A downward force was applied on the pin holes shown in Fig. 3.8(a).

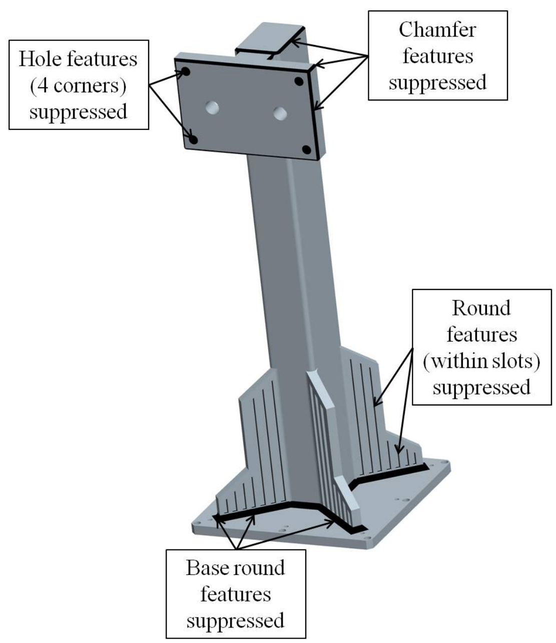
3. **Part-C**: The constraints are located at the rectangular slots in the model that are connection points to a vertical piece. A distributed vertical load was placed on the plane of the face perpendicular to the vertical plane (Fig. 3.9(a)).

4. **Part-D**: The constraint was applied at the location of the hole, where this part will be bolted into place. Multiple forces were applied at different locations of the model, (1) tensile forces were applied at the hole on the free end and (2) another force was applied on the other end of the part in the opposite direction at the middle pin shown in Fig. 3.10(a).

5. **Part-E**: The part slides into grooves on a wall support and then pinned into place. The top curve latches on a lip and acts as the constraint of the model. A force was applied in the vertical direction on the horizontal face (Fig. 3.11(a))

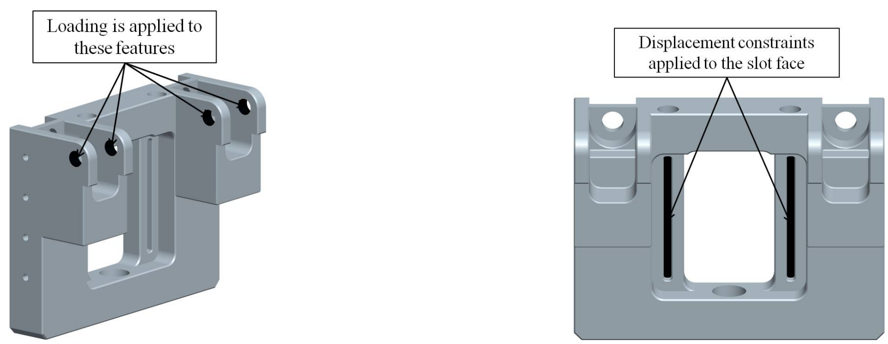


(a) Boundary conditions

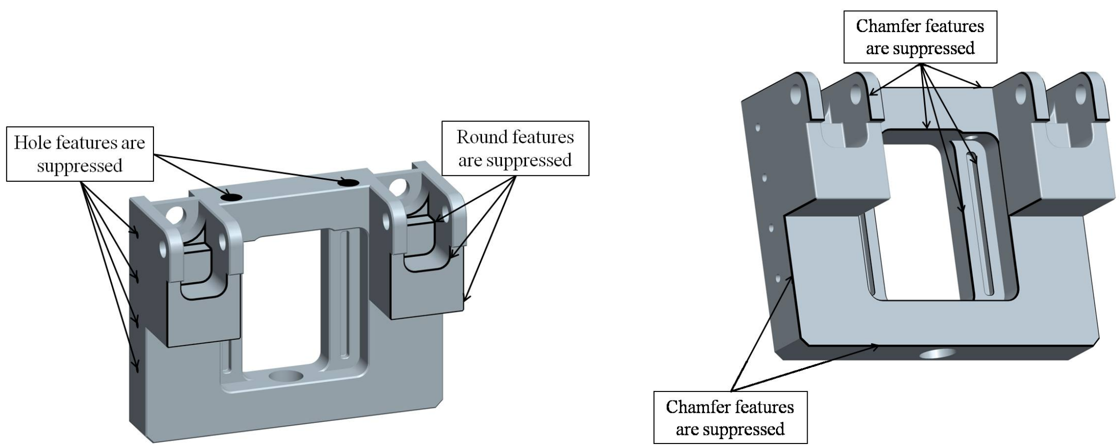


(b) Suppression of features

Figure 3.7: Part A: boundary condition features and feature suppression after rule implementation

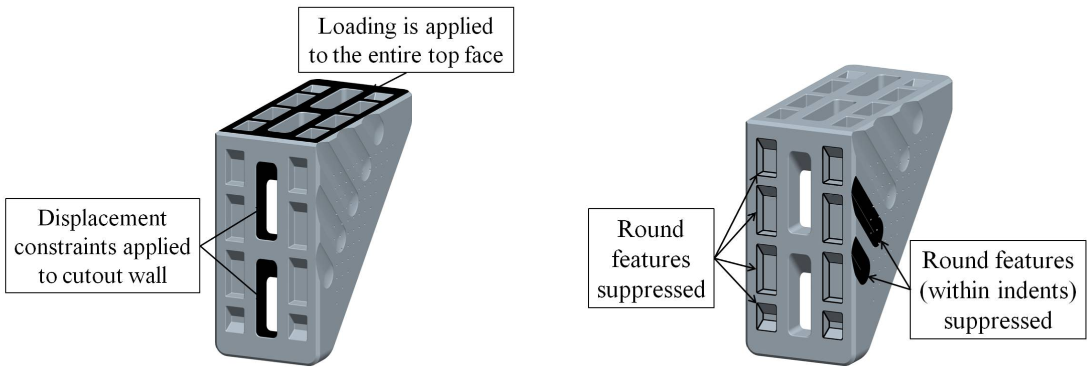


(a) Boundary conditions

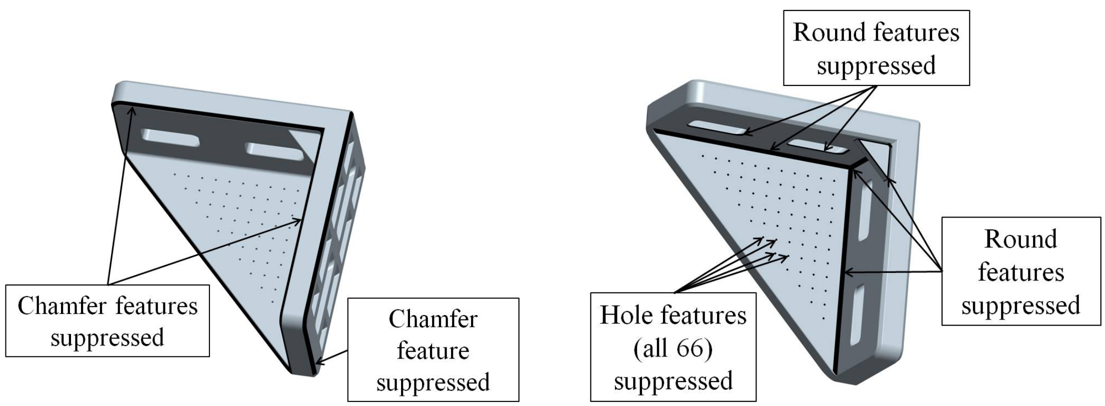


(b) Suppression of features

Figure 3.8: Part B: boundary condition features and feature suppression after rule implementation

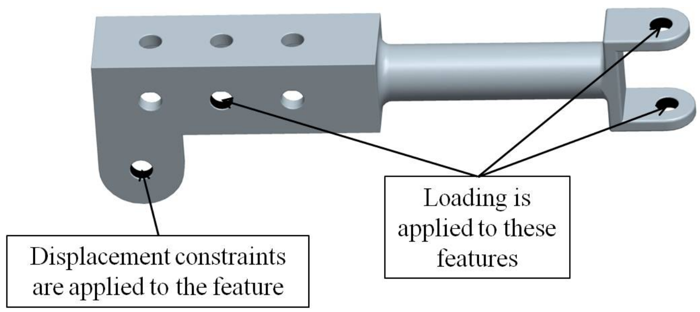


(a) Boundary conditions

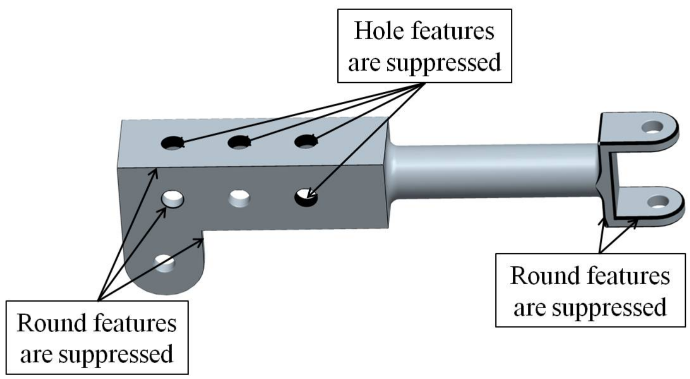


(b) Suppression of features

Figure 3.9: Part C: boundary condition features and feature suppression after rule implementation

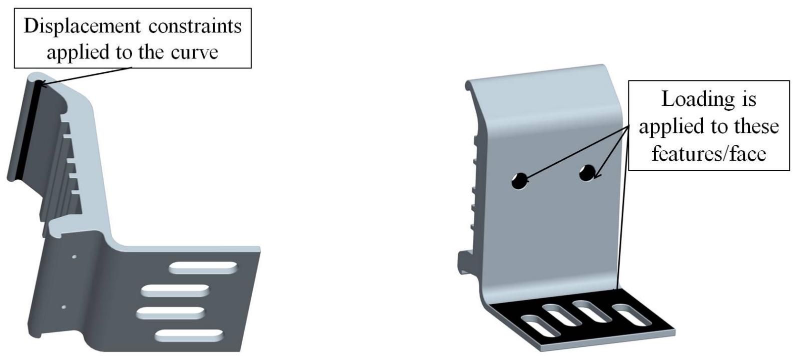


(a) Boundary conditions

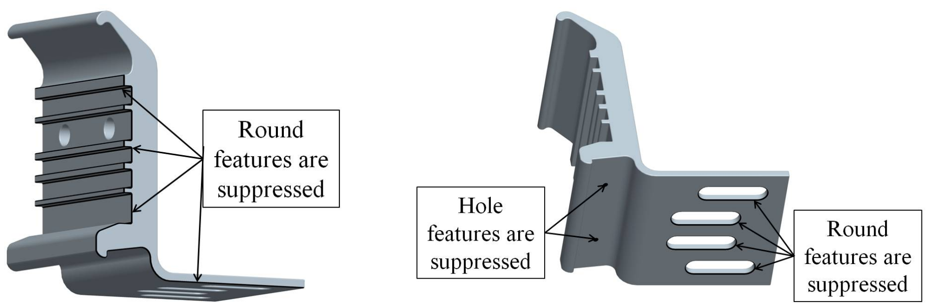


(b) Suppression of features

Figure 3.10: Part D: boundary condition features and feature suppression after rule implementation



(a) Boundary conditions



(b) Suppression of features

Figure 3.11: Part E: boundary condition features and feature suppression after rule implementation

Now, an analysis was conducted independently for both the original and simplified models of these parts after applying the boundary conditions described above. First the learned rules were applied to the five test parts. After applying the learned rules, the suppressibility_state of the features by the learned rules were compared to the suppressibility_states of the features given by an expert analyst. The learned rules proved to be highly accurate for the five test parts compared to the expert analysis of the parts. For the five parts there were a total of two feature classification errors of the total 154 features in the models, both arising within the round features. The errors were caused due to rare suppression occurrences within the simplification method, but found to be conservative in nature. One round was relatively larger and the other was relatively closer to a boundary condition when compared to the standard values for round suppression. Table 3.7 displays the results of the rules to the actual suppression analysis.

Meshes were then generated for these parts in Pro/Mechanica to allow for a comparison of analysis results in terms of mesh quality, mesh generation time, and maximum Von Mises stress between the original and simplified models. The results shown in Table 3.8 demonstrate a significant reduction in the amount of elements by more than a factor of two, with the largest being a factor of over eight. There was also a decrease in the maximum aspect ratio, signifying a higher quality mesh. Due to the simplification, the time to mesh was also greatly reduced by at least 27 seconds; this could be significant when analyzing numerous parts.

To evaluate the effectiveness of the simplification process, the maximum Von Mises stresses were compared between the original and simplified models. In each

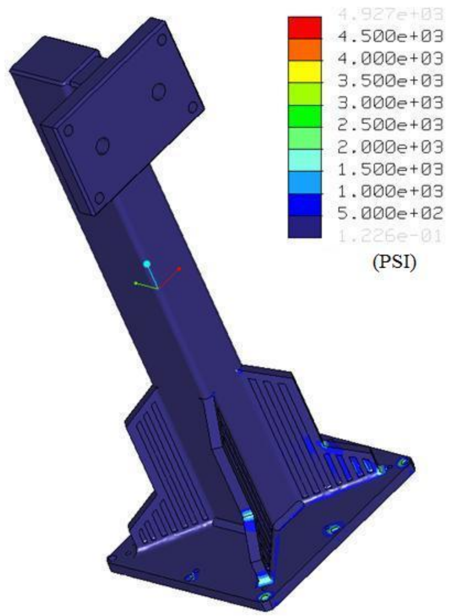
Feature Type	**No. of Features**	**No. of Errors**	**Error Type**
Hole	93	0	None
Round	44	2	Conservative
Chamfer	17	0	None

Table 3.7: Analysis of learned rules on the five test parts (A)-(E).

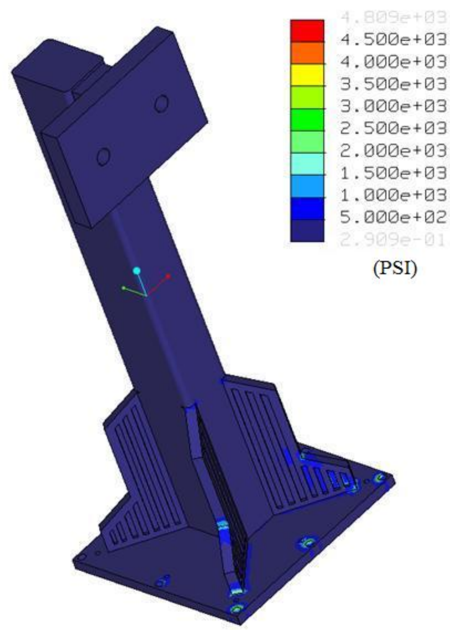
case there is a difference of less than five percent, indicating that the simplification process did not significantly alter the accuracy of the results. The stress analyses results for both the unsimplified and simplified models can be seen in Figs. 3.13-3.16. In the results of part C, the critical stress values are displayed to show that this location is constant between the original and simplified models.

	Tetra Elements	**Max. Aspect Ratio**	**Meshing Time (sec)**	**Max. Von Mises Stress (MPa)**
Part-A original	19500	11.39	81	33.97
Part-A simplified	4511	11.20	7.2	33.15
Part-B original	6848	20.09	22.2	372.39
Part-B simplified	2462	11.20	7.8	373.01
Part-C original	20643	20.34	165	30.06
Part-C simplified	7865	15.60	46.9	29.93
Part-D original	7706	11.17	28.2	42.27
Part-D simplified	1445	10.59	4.8	43.61
Part-E original	9164	11.45	43.2	162.30
Part-E simplified	4049	11.19	13.8	159.82

Table 3.8: Analysis of models before and after simplification

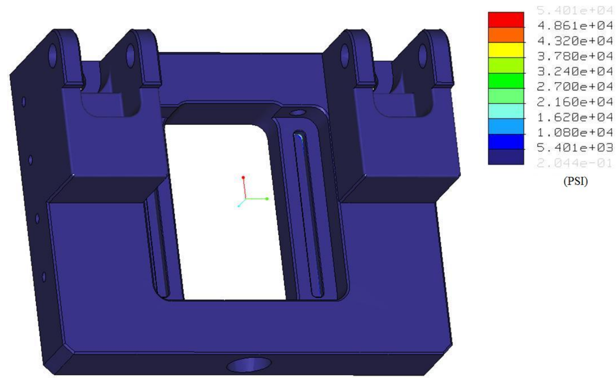


(a) Unsimplified model

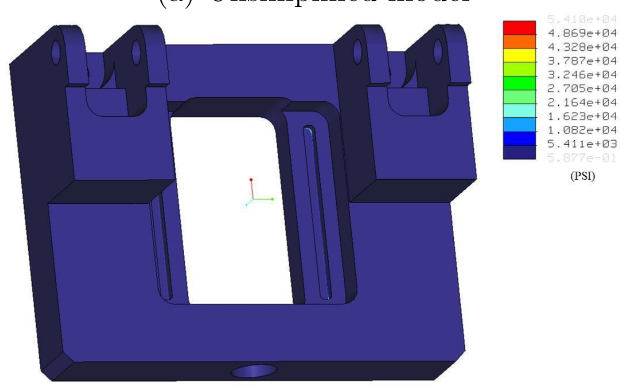


(b) Simplified model

Figure 3.12: Stress analysis results for the unsimplified and simplified models for part A

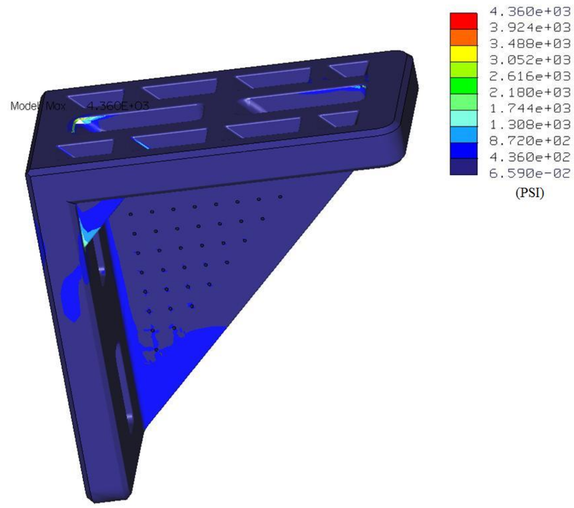


(a) Unsimplified model

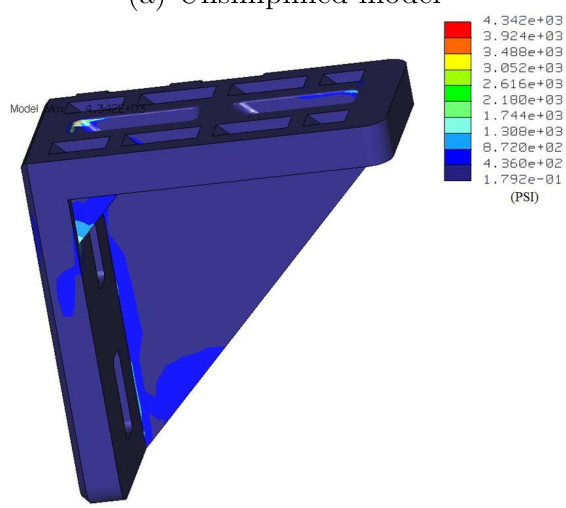


(b) Simplified model

Figure 3.13: Stress analyses results for the unsimplified and simplified models for part B

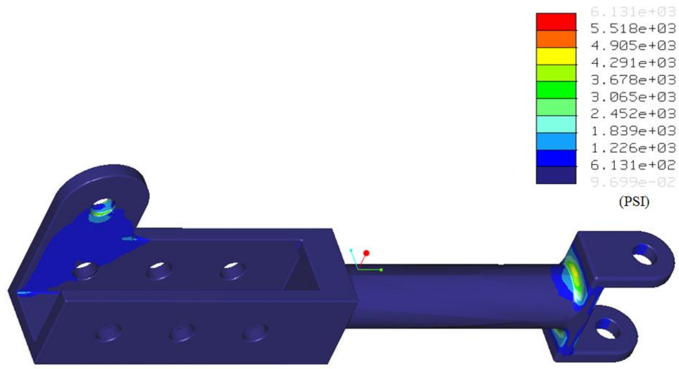


(a) Unsimplified model

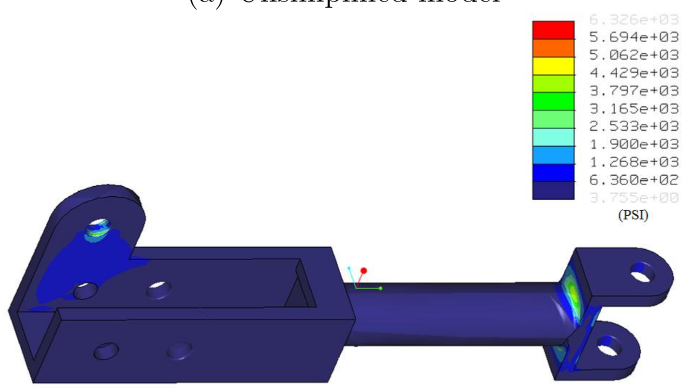


(b) Simplified model

Figure 3.14: Stress analyses results for the unsimplified and simplified models for part C

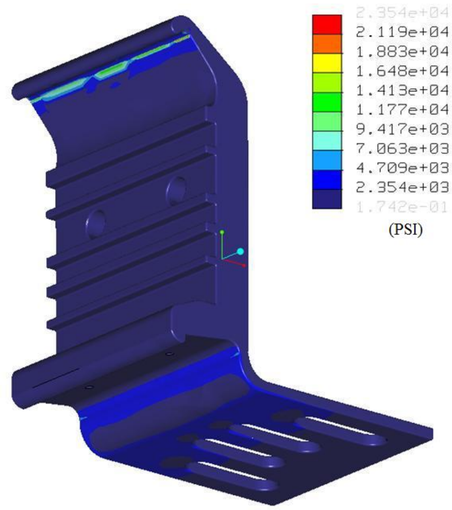


(a) Unsimplified model

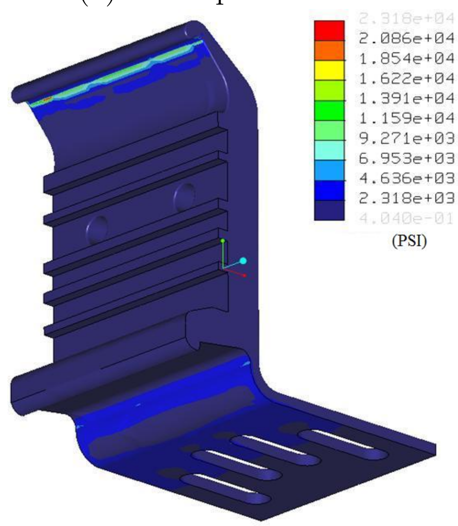


(b) Simplified model

Figure 3.15: Stress analyses results for the unsimplified and simplified models for part D



(a) Unsimplified model

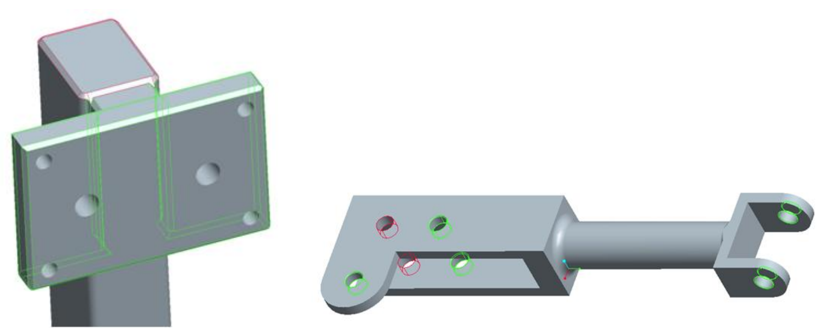


(b) Simplified model

Figure 3.16: Stress analyses results for the unsimplified and simplified models for part E

After conducting finite element analysis, the five simplified parts showed acceptable errors within 5% and improved mesh quality when compared with the original models. The learned rules proved to be highly accurate for the test parts. As mentioned earlier only conservative errors were allowed for within the analysis process.

Examination of how the parent-child relationships affect the results was done. For that the parts A and D were considered, shown in Fig. 3.17(a)-(b). Within these models the parent child dependencies were accessed and there were features that needed to be retained in order to preserve parent-child relationships. The chamfers shown in red in Fig. 3.17(a) has its children shown in green. If the parent is suppressed then these children will also get suppressed. However, in this case the children also meet the suppression criteria, hence both the parent and child would be suppressed. The holes shown in the Fig. 3.17(b) cannot be suppressed because the children are labeled `BOUNDARY_CONDITION`.



(a) Part-A (b) Part-D

Figure 3.17: Example of suppression based on parent-child relationships

3.5 Summary

In this chapter, a framework was presented that can be utilized in any of the existing CAD softwares for suppressing geometric features before conducting finite element analysis. The automated tool developed in this chapter assists users in identifying the suppressible features for FEA and it reduces the burden on users in identifying suppressible features manually. Towards generating this automated tool, initially a rule language was developed that can capture experts' suppression knowledge. This knowledge was acquired through expert demonstrations and the rules were expressed in terms of a decision tree. With the help of statistical induction learning techniques suppression rules were learned for suppressing holes, rounds and chamfers.

The Pro/Toolkit API available in ProE was utilized to generate a model simplification tool; for hole, round and chamfer features. The complete model is traversed through to determine the features that were capable of suppression based upon the rules learned through expert demonstrations. The suppression state was determined by the feature type, dimensions, and relation to other features, i.e. parent-child relationship and inter-feature distances. To demonstrate the simplification method, five different models were analyzed. Evaluation of the learned rules found two out of 154 features that should have been suppressed, but were given the suppressibility_state of FALSE; this error was of conservative nature. In comparison to the original models, the simplified models had a decrease of at least two percent in the amount of elements within the mesh. Furthermore, the maximum Von Mises stress

was within the error range of 5%.

Chapter 4

Assembly Model Simplification for Finite Element Analysis

4.1 Overview

The main focus of the assembly approach is to design a framework that assists the user in the simplification of complex CAD assembly models, consisting of numerous part models; similar to the part simplification framework. The overall approach can be broken down into a process flow, displayed in Fig. 4.1. Initially, the complete assembly information is extracted and written in an XML format. This information is stored as an element tree that contains all information about the parts, including, for example: all attributes, such as the material, and all references to the existing parts, such as placement and orientation. Upon extraction of the model information, the insignificant parts are then determined and the model is simplified, whether parts are suppressed or replaced.

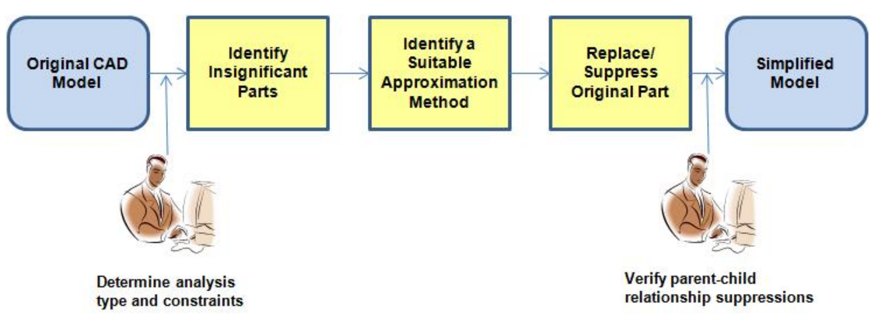


Figure 4.1: Overview of the assembly approach

Discussions with AVMI engineers introduced three key parameters that can be used to assist in the classification of parts as structurally significant or insignificant. These parameters include DESCRIP, PART_NAME, and MATERIAL.

1. The DESCRIP parameter often provides part information. This information can include the type of the part; e.g. nut, washer, or screw (standard hardware), and characteristics of the part type, e.g. thread pitch, bolt diameter, and head shape (bolt or screw).

2. The generation of the PART_NAME is usually standardized throughout the industry, grouping similar parts. For instance, within AVMI a PART_NAME including the AN*, MS*, or NAS* string is standard hardware, whereas a PART_NAME including the PE* or #####M* string is a unique part.

3. The part material is dependent upon the requirements of the part within the overall assembly. Different materials are used for different types of parts, so a specific material can be seen as insignificant within the overall model.

A search within the XML file for key words and/or numbering schemes of the specified parameters can indicate whether or not the part is standard hardware or a unique component. Standard hardware is generally not advantageous within the FEA model, as they generate unnecessary mesh complexities within the analysis and can be suppressed; whereas a unique component is a designed part that requires FEA, but may be capable of simplification.

The framework has been implemented within the ProE CAD system. Section 4.2 describes the algorithm for suppressing parts based upon the PART_NAME and

MATERIAL and usefulness of this technique. Section 4.3 describes simplification methods pertaining to the individual parts of the assembly. Section 4.4 presents results of the simplification process, by using the techniques described in the previous sections.

4.2 Part Suppression

In order to assist in the simplification process of the assembly models, the Pro/Toolkit API in ProE was utilized to generate a model suppression tool, just as in the part simplification framework. For recap, this tool is an add-on module in ProE, accessible through the menu systems in the ProE environment. In the API, the design model is stored by ProMdl, a data handle to retrieve the complete information of the design model. This information is stored within ProE as an element tree that contains all information about the parts, including, for example: all attributes, such as the material, and all references to the existing parts, such as placement and orientation. Suppression is based upon the part MATERIAL and PART_NAME parameters.

4.2.1 Part Utilization

Defining the intended purpose of the part within the overall assembly model; e.g. structural stability, thermal enhancement, ergonomic aspect, or part attachment method; can facilitate the classification of the part's criticality for analysis. The PART_NAME and DESCRIP parameters can aid in the identification of the

part's purpose within the assembly. Generally, the PART_NAME has a common string, specifying the group type to which the part belongs; for example the MS* string refers to a type of fastener and then the DESCRIP parameter contains key words to assist in specifying the intended purpose, such as bolt or screw.

Although this can facilitate the classification, the intended purpose of the part is not always clearly defined. Different aspects of the part can be intended for various purposes or parts unintended for the specified purpose can have an effect on different analysis results; e.g. a heat shield that is placed within an assembly for the thermal protection of other parts, may not only affect a thermal analysis (for which the part is intended), but could also have an effect on the modal analysis of the overall model. For these reasons, focus was geared toward a specific part type that mainly falls into the intended purpose of attachment methods: hardware (i.e. fasteners), such as nuts and bolts. Depending on the number of parts in the assembly, a model could potentially contain a couple hundred fasteners. Even with the large amount of fasteners, there is often only a select few of different fasteners within the model. Using the same fasteners throughout reduces the overall part count of the model. There are several benefits to reducing the overall part count of an assembly; (1)decreases some of the cost because less material types need to be purchased, (2) similar tools can be used to assemble the model and (3) reduces the error probability during the manufacturing process of the model. Typically, these fasteners have minimal to no effect on analyses and, therefore, fasteners are frequently suppressed before an analysis; greatly reducing the computational cost of the analysis.

Currently, an engineer will have to manually transverse through the model and/or the model tree and independently suppress each fastener. With a large model this could become a tedious time consuming task. To help expedite the process of fastener suppression, or other parts that may appear several times within the model, the framework can suppress the parts by the PART_NAME; all parts in the model with that part name will be quickly suppressed. For instance, a model containing 25 bolts of two different types (PART_NAMEs), would only require two independent suppressions, instead of the manual independent suppressions of 25.

4.2.2 Material

In addition to the part's intended purpose, material selection plays a role in determining the extent of the part criticality, from the analysis results point of view. The analysis response of a part is highly dependent upon its overall shape and dimensions, but the magnitude of this response is a result of the material; i.e. the material properties. For instance, Fig. 4.2 displays an example assembly model section, which is the input for two static analyses. For comparison the parts in red are constucted of a similar material and only the material of these two parts is altered between the two tests. The material properties of the two parts for the analyses are that of steel and nylon. From the results shown in Figs. 4.3 and 4.4, steel and nylon material attributes respectively, it can be deduced that the critical stress within the model is much higher in the assembly model where the parts are made of steel, than that of the assembly model where the parts are made of nylon. This stress

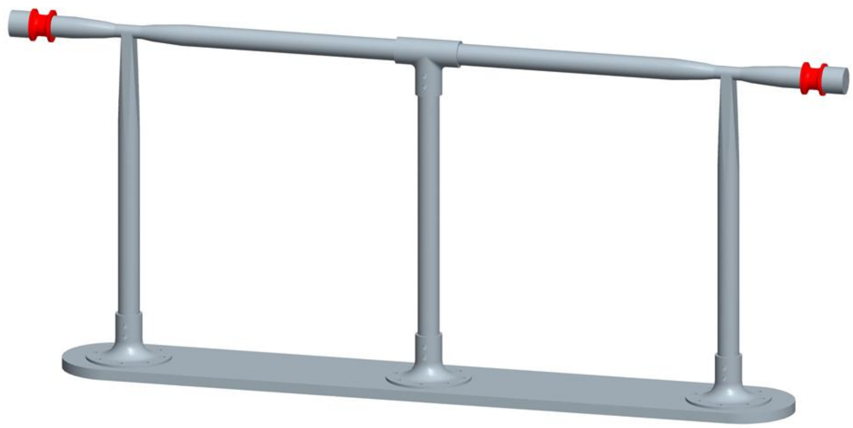


Figure 4.2: Example assembly model

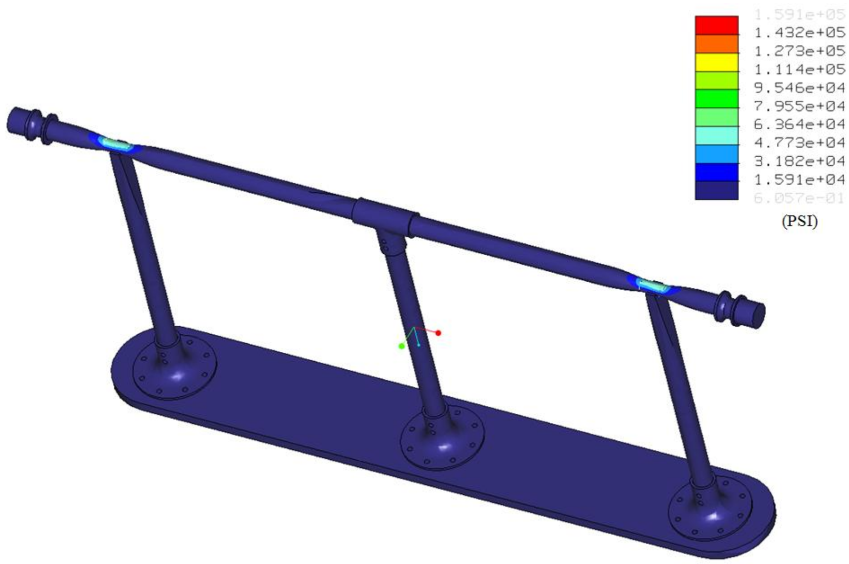


Figure 4.3: Stresses within the model when the two parts are made of the material steel

difference is the result of the difference in the densities of the two materials. The analysis demonstrates that the part material is critical in determining the criticality of the part for analysis purposes.

The behavioral response of the material, due to its properties as discussed

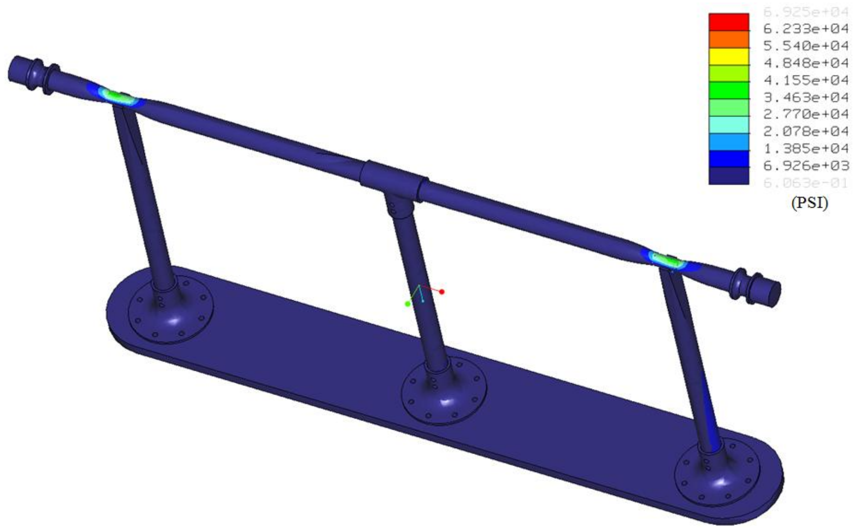


Figure 4.4: Stresses within the model when the two parts are made of the material nylon

above, has generated basic standards within the material selection phase; for instance, a load bearing part is generally constructed of a material with a higher strength, such as a metal compared to a plastic. Knowledge of these standards and the analysis type to be performed on the model can signify material types as insignificant or significant to the analysis. For example, in a structural analysis, often, plastic parts can be viewed as insignificant, since load bearing parts are usually not constructed with this material.

4.2.3 Algorithm

The parts considered to be insignificant based upon their material or purpose are then suppressed. Either the MATERIAL or PART_NAME value can be entered to suppress all the parts containing that parameter value. Before suppression it

is necessary to understand the dependency relationships among the parts, just as discussed in the previous chapter. The previous chapter discusses the feature-based approach in model generation, where features are dependent among one another within a model; generating the feature history tree [33, 37]. Just as the features are dependent upon one another in a part model, parts are dependent upon one another within an assembly model. When constructing the assembly model, parts are introduced into the model one at a time referencing the previous parts within the model for placement. The same model can be constructed in several ways , dependent upon the order in which the parts were added to the model, generating different history trees and, ultimately, different dependencies among the parts. Suppression of a parent will ultimately suppress its children parts, hence, it is imperative to understand parts' parent-child relationships for suppressions; since the child could potentially be a critical part for the analysis.

For example, let's look at the construction of a simple table with four legs, attached by nuts and bolts, illustrated in Fig. 4.5. Conventionally the table top would be first inserted into the model. Then the legs would be brought in one at a time, referencing the axis of the holes within the table top for the desired location of the legs. Upon including the legs, the bolts would be inserted into the table top and legs by referencing the axes once again to line up the bolts and then referencing the table top surface and connecting surface of the bolt head. After this the nuts are added by referencing the bolt axis and the connecting surfaces of the nut and table leg. In this instance, suppression of the bolts would also suppress the nuts, since they are referenced from the bolts axis. This setup would not cause any critical features

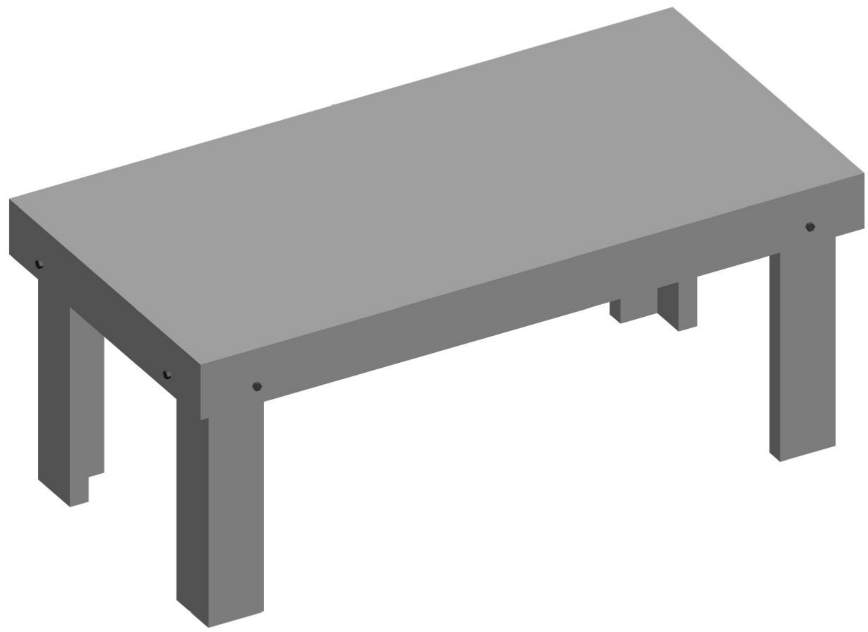


Figure 4.5: Simple four-legged table for illustration of the parent-child relationship within a model

to be suppressed within the model, however, a designer could assemble the table in a different manner within the software. Let's say the designer unconventionally decides to bring the bolts into the model and reference the table top and legs from the bolts. Then suppression of the bolts would ultimately suppress the entire model. Extracting these parent-child dependencies is crucial in order to avoid suppressing any non-critical features.

An algorithm has been developed to account for the parent-child relationships during the suppression process, similar to that of the algorithm presented in the previous chapter for suppressing features based upon the parent-child relationships. Alteration of the algorithm was required, since there is not a general rule set that assigns a suppressibility status based upon all the criteria; i.e. suppression is based

upon either the MATERIAL or PART_NAME value.

The algorithm for suppressing features is as follows:

1. Construct G (parent-child relationship graph) by inserting a node V corresponding to every part in the tree, which is not suppressed. If part p' uses a part p as a reference, then insert an edge (v, p') in G. Here the vertex v corresponds to part p and vertex v' corresponds to part p'. Fig. 4.6 shows an example of the parent-child relationship graph.

2. Find nodes X in V that do not have children and have the attribute value (MATERIAL or PART_NAME) given. If X is empty proceed to step 3, otherwise do the following:

   (a) Remove every node in X from G by deleting the associated vertex and edges.

   (b) Go to step 2.

3. Find nodes Y in V that have children and the attribute value and do the following:

   (a) For node Yi, highlight the child/children in green and the parent node in red, ask if the user would like to suppress the parent and associated child/children. If Yi is empty, then stop.

      i. Yes, then remove node Yi and child/children nodes of G by deleting the associated vertex and edges. Proceed back to step 3a.

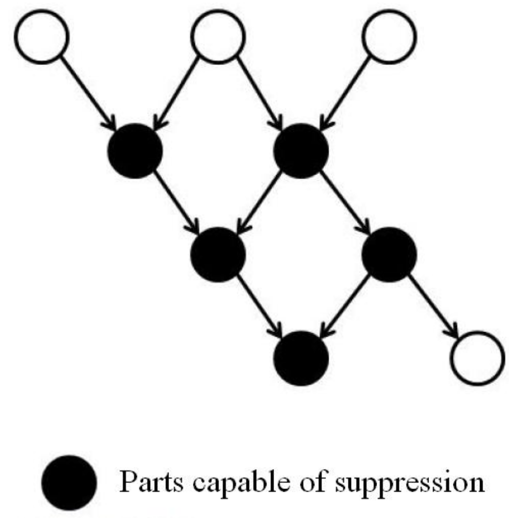


Figure 4.6: Example of the generated parent-child relationship graph

ii. No, then add 1 to i and proceed to step 3a.

This model is only valid for assembly models generated using only parts and no minor assembly models. When minor assembly models are used within the overall assembly, the parent references of the minor assembly are not necessarily used as the reference for inclusion into the overall model. If the parent is not used, then it is not seen as a child to the parts from which the minor assembly is referenced. Then suppression of the part model of the overall assembly used for referencing generates an error because the parents of the referenced part are not suppressed, but the references for the part into the overall model are suppressed.

4.3 Simplification

Suppression of parts is not the only means of simplifying an assembly model, the parts themselves can also be simplified within the model. Simplification of the

parts can be accomplished by suppressing the non-critical features, or the parts can be replaced, either with a simpler geometry or point mass that maintains the effects of the original part.

The first method was thoroughly discussed in Chapter 2. To successfully use this approach within ProE, the simplification of the part must be done within the part file because modifications cannot be done to an individual part within the assembly file. Even though the modifications, such as suppression of features, are done within the part file, they will be automatically reflected within the assembly model.

The second method requires additional knowledge compared to that of the other method, since the features are not only suppressed, but the part geometry is altered; maintaining a similar response to that of the original part. First, the analysis type must be identified, since a part can have a different response within the various types of analyses. Upon knowledge of the analysis type, the part must be quickly reviewed to determine the necessity and response, so a suitable approximation method can be developed, whether the method is point mass or simpler geometry with density match. Then the center of gravity (CG) is calculated of the original part. For the point mass simplification the CG is the location of the point mass. For the simpler geometry simplification the simpler geometry is designed to maintain the location of the CG. Figure X illustrates a part that was simplified to a simpler geometry displayed in Figure Y. To automate this method an outside program requiring numerous inputs will be needed to analyze and create a simpler geometry according to the model.

Figure 4.7: Unsimplified boom model for a modal analysis

4.4 Results and Discussion

To illustrate the framework described, assembly models have been simplified for both a structural and a modal analysis. Simplification of the models was accomplished by part suppression, based upon the MATERIAL or PART_NAME attribute value, and part simplification discussed in the previous chapter. Upon model simplification a finite element analysis is conducted for both the simplified and original models, with the help of Pro/Mechanica. After conducting the FEA, the analysis results were compared between the simplified and original models in terms of mesh quality(number of tetra elements and maximum aspect ratio), mesh generation time, and either mode values or maximum Von Mises stress, depending on the analysis type.

4.4.1 Modal Analysis

A modal analysis was done on a boom assembly model shown in Fig. 4.7. The simplification of the assembly model, displayed in Fig. 4.8, was done in the following manner:

- The 15 bolts and their respective nuts were suppressed by their PART_NAME.

Figure 4.8: Simplified boom model for a modal analysis

To suppress the 15 nuts it only took two different PART_NAMEs, which accounted for the two different diameter nuts; although it required four PART_NAMEs to suppress the bolts, accounting for the different lengths and diameters. To complete the suppression of the 30 parts it only took 6 passes compared to the manual suppression of 30 passes. For clarification the nuts were suppressed prior to the bolts, removing the children parts before the parent parts. If the bolts were first suppressed, then the nuts would have been seen as children to the bolts and the suppression message would have appeared. Either order of suppression achieves the same desired outcome.

- The two mount attachments, the center slider, the complex attachment, and the boom were simplified using the method discussed in the previous chapter.

The first step in the FEA process is mesh generation. The comparison of the mesh quality and mesh generation time between the two models can be seen in Table 4.1. As illustrated below the simplified model has a 44% reduction of the number of elements; the max aspect ratio is reduced by 7.75, which means the mesh quality is better in the simplified model; and the time to mesh was reduced by almost 20% or 13.8 sec. In terms of the mesh, the simplified model has a better quality mesh and was generated faster than the original model.

	Tetra Elements	Max. Aspect Ratio	Meshing Time (sec)
Original	15069	20.90	72
Simplified	8422	13.15	58.2

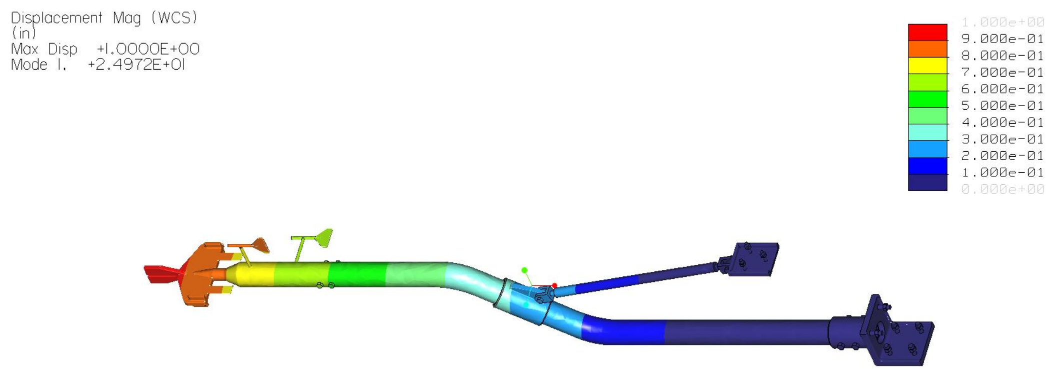
Table 4.1: Mesh improvement

Mode	Original Model (Hz)	Simplified Model (Hz)	% Error
1	25.23	25.28	0.1982
2	73.50	73.47	0.0408
3	75.32	74.75	0.7568
4	77.21	77.16	0.0648

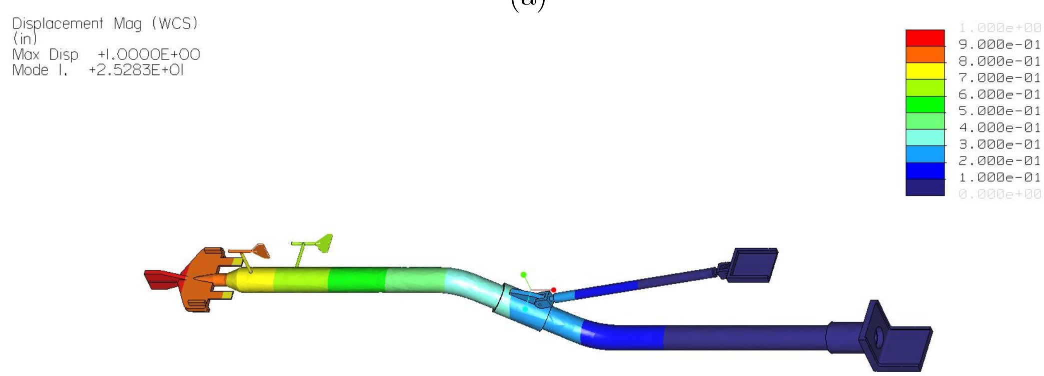
Table 4.2: Modes calculated within the modal analysis

The mesh quality is not the only factor required for comparison of the models, but the analysis results of the model must be similar or the simplified model cannot replace the original model within the analysis process. A modal analysis has been executed on the boom models and the first four modes of both the original and simplified models were compared; shown in Table 4.2. The last column of the table shows that there was an error of less than 1% between the two models for all four of the modes.

The modes of the two models can be seen in Figs. 4.9-4.12.

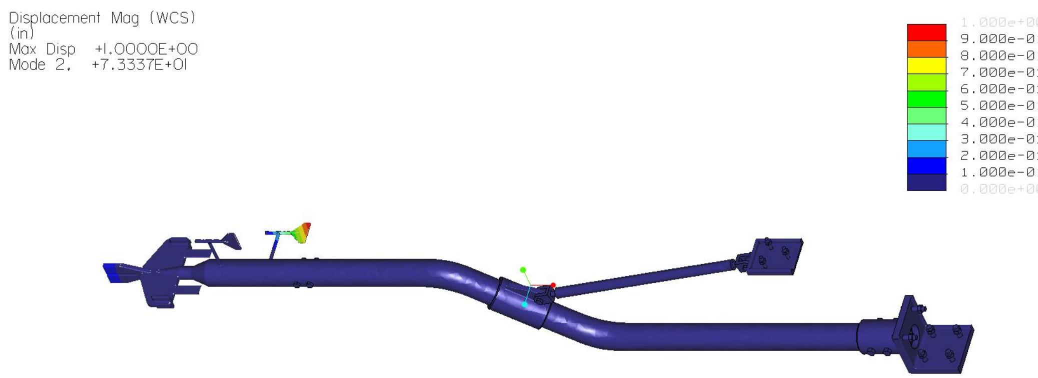


(a)

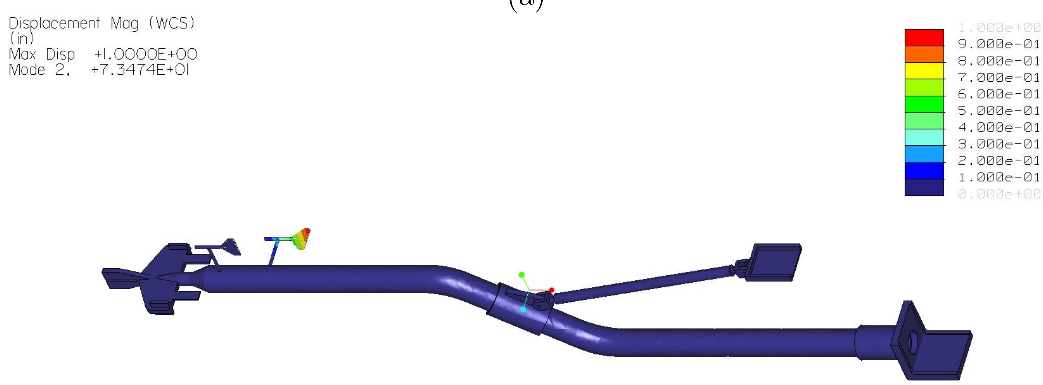


(b)

Figure 4.9: (a) Mode 1 of the unsimplified boom assembly (b) Mode 1 of the simplified boom assembly

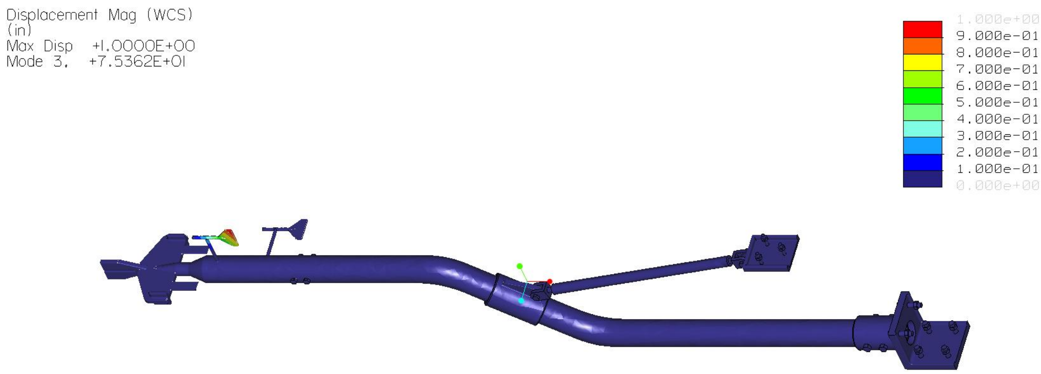


(a)

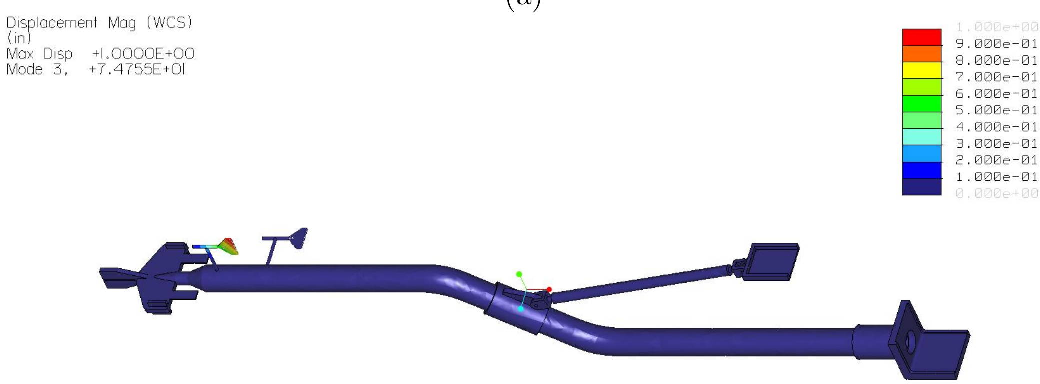


(b)

Figure 4.10: (a) Mode 2 of the unsimplified boom assembly (b) Mode 2 of the simplified boom assembly

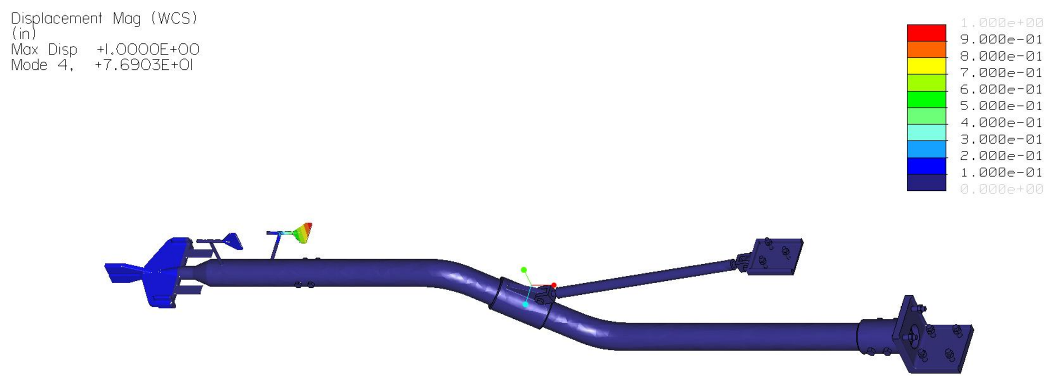


(a)

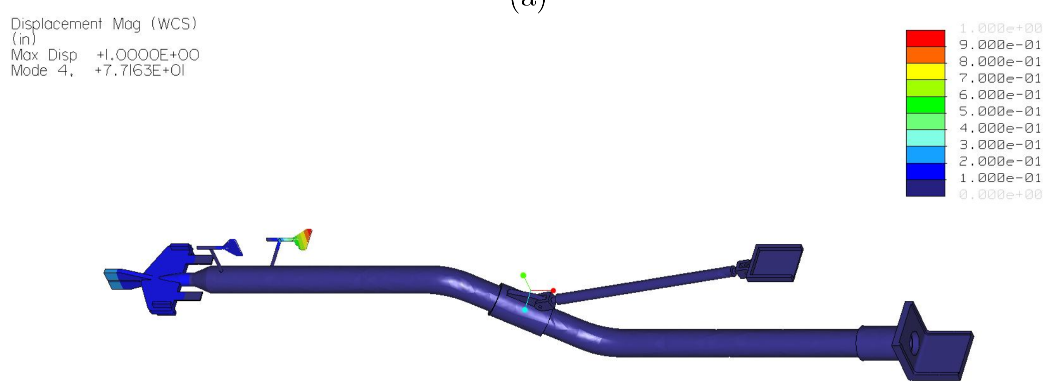


(b)

Figure 4.11: (a) Mode 3 of the unsimplified boom assembly (b) Mode 3 of the simplified boom assembly



(a)



(b)

Figure 4.12: (a) Mode 4 of the unsimplified boom assembly (b) Mode 4 of the simplified boom assembly

4.4.2 Structural Analysis

In a similar fashion to the modal analysis, a structural analysis was completed for two assembly models. The two models used for the structural analysis were a weight bench assembly and the same boom assembly from which the modal analysis was done. Comparison between the simplification process for the modal analysis and structural analysis of the beam will highlight different simplification processes.

First, the original and simplified models of the weight bench are shown in Figs. 4.13 and 4.14, respectively. Simplification of the assembly was done in the following manner:

Figure 4.13: Unsimplified weight bench assembly model for a structural analysis

- The 12 bolts and their respective nuts were suppressed by their PART_NAME. To suppress the 12 nuts it only took two different PART_NAMEs, which accounted for the two different diameter nuts; although it required three

Figure 4.14: Simplified weight bench assembly model for a structural analysis

PART_NAMEs to suppress the bolts, accounting for the different lengths and diameters. To complete the suppression of the 24 parts it only took five passes compared to the manual suppression of 24 passes. For clarification the nuts were suppressed prior to the bolts, removing the children parts before the parent parts. If the bolts were first suppressed, then the nuts would have been seen as children to the bolts and the suppression message would have appeared. Either order of suppression achieves the same desired outcome.

- The support frames were simplified using the method discussed in the previous chapter.

- The attachments covering the ends of the pipes are all constructed of a nylon material and have no bearing on the stresses generated within the model. These 9 caps were suppressed in one pass, using their MATERIAL value.

Just as in the modal analysis, the first step in the structural analysis is mesh generation. The comparison of the mesh quality and mesh generation time between the two models can be seen in Table 4.3. As illustrated below the simplified model has a 35% reduction of the number of elements; the max aspect ratio is reduced by 2.27, which means the mesh quality is better in the simplified model; amd the time to mesh was reduced by about 38% or 150.68 sec. In terms of the mesh, the simplified model has a better quality mesh and was generated faster than the original model. Upon mesh generation the Von Mises stresses are calculated within the model and the maximum stress is then compared between the two models, shown in Table 4.3, which has an error of 4.8% between the original and simplified weight bench models. Figs. 4.15 and 4.16 show the stresses within the two models

	Tetra Elements	**Maximum Aspect Ratio**	**Meshing Time (sec)**	**Max. Von Mises Stress**
Original	46423	19.78	391.8	39.93
Simplified	30133	17.51	241.2	41.85

Table 4.3: Mesh improvement

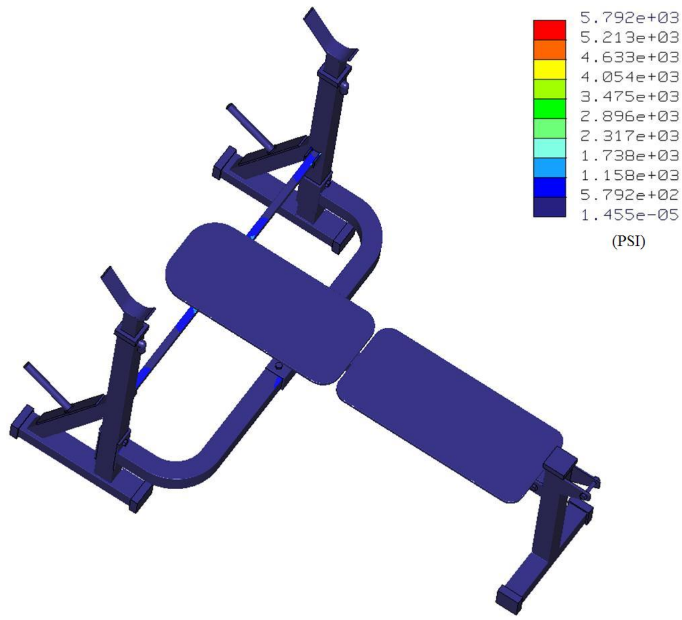


Figure 4.15: Von Mises Stress distribution on the unsimplified weight bench assembly

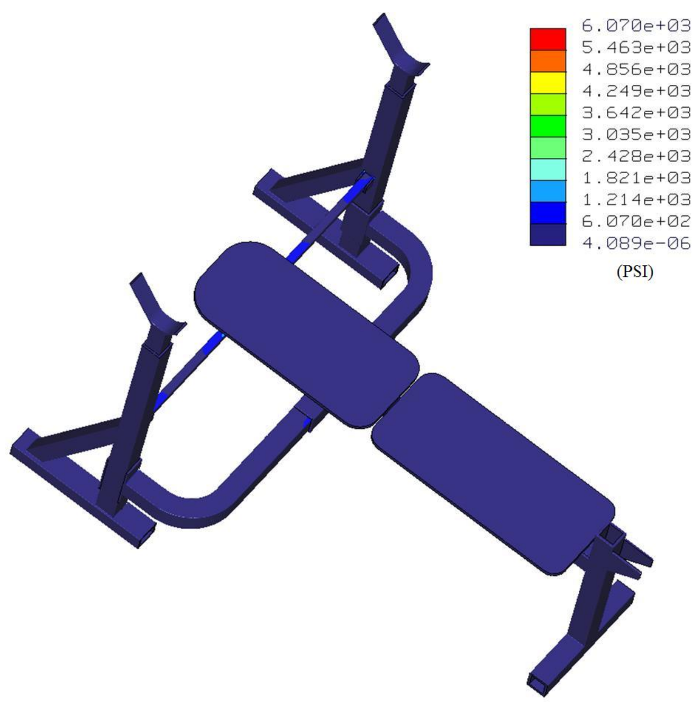


Figure 4.16: Von Mises Stress distribution on the simplified weight bench assembly

The other assembly model for which a sructural analysis was done was the boom assembly. Figs. 4.17 and 4.18 illustrate the original and simplified models of the boom assembly. The original model, Fig. 4.17, is the same as the original modal analysis model, Fig. 4.7, but the simplified models are slightly altered. The structural analysis simplified model has replaced the complex structure on the end of the boom with a simpler part; where in the modal analysis this complex end was not replaced, but just simplified, according to the previous chapter. Simplification of the assembly was done in the following manner:

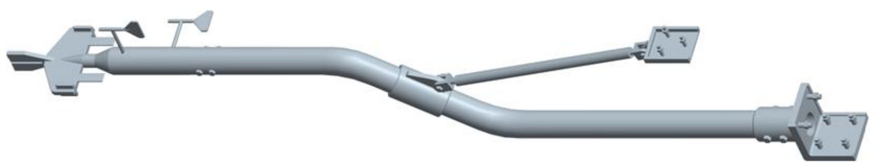


Figure 4.17: Unsimplified boom assembly model for a structural analysis

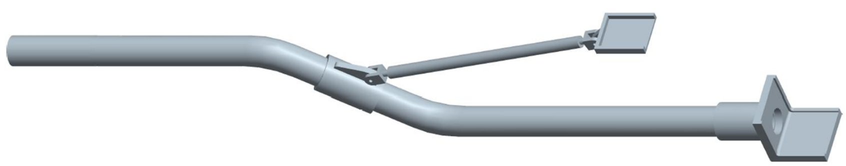


Figure 4.18: Simplified boom assembly model for a structural analysis

- The 15 bolts and their respective nuts were suppressed by their PART_NAME. To suppress the 15 nuts it only took two different PART_NAMEs, which accounted for the two different diameter nuts; although it required four PART_NAMEs to suppress the bolts, accounting for the different lengths and diameters. To complete the suppression of the 30 parts it only took 6 passes compared to the

manual suppression of 30 passes. For clarification the nuts were suppressed prior to the bolts, removing the children parts before the parent parts. If the bolts were first suppressed, then the nuts would have been seen as children to the bolts and the suppression message would have appeared. Either order of suppression achieves the same desired outcome.

- The two mount attachments, the center slider, and the boom were simplified using the method discussed in the previous chapter.
- The complex attachment at the end was replaced by a simpler part geometry. The complexity of the part was reduced to a cylindrical geometry, maintaining the CG of the part. This was not capable within the modal analysis because the as the mode number increased there was a higher emphasis on the mode value from the complex features of the attachment.

The comparison of the mesh quality and mesh generation time between the two models can be seen in Table 4.4. As illustrated below the simplified model has a 35% reduction of the number of elements; the max aspect ratio is reduced by 9.23, which means the mesh quality is better in the simplified model; and the time to mesh was reduced by about 69.2% or 49.8 sec. In terms of the mesh, the simplified model has a better quality mesh and was generated faster than the original model. Upon mesh generation the Von Mises stresses are calculated within the model and the maximum stress is then compared between the two models, shown in Table 4.4, which has an error of 4.8% between the original and simplified boom models. Figs. 4.19 and 4.20 show the stresses within the two models.

	Tetra Elements	**Maximum Aspect Ratio**	**Meshing Time (sec)**	**Max. Von Mises Stress**
Original	15069	20.90	72	101.49
Simplified	5141	11.67	22.2	96.59

Table 4.4: Mesh improvement

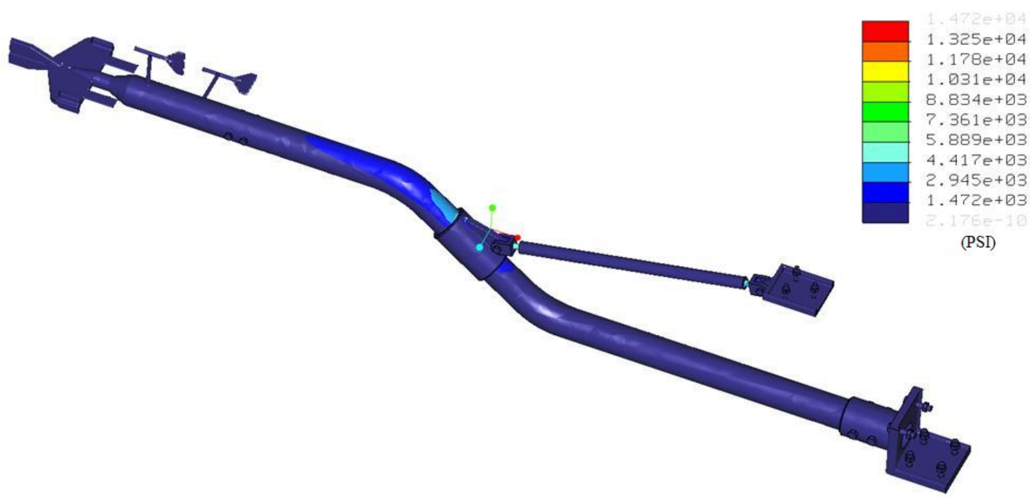


Figure 4.19: Von Mises Stress distribution on the unsimplified boom assembly

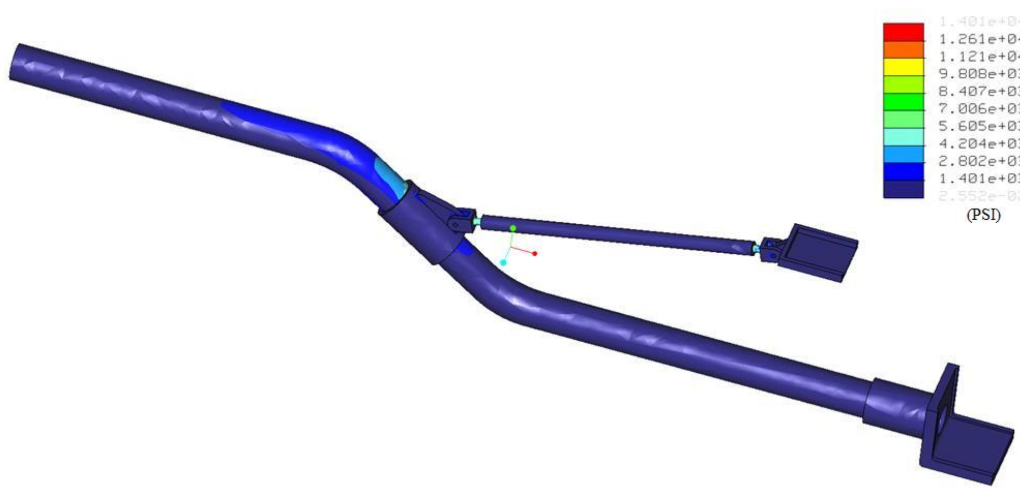


Figure 4.20: Von Mises Stress distribution on the unsimplified boom assembly

4.5 Summary

This chapter has presented a tool to be used for the simplification of an assembly model for FEA, utilizing the Pro/Toolkit available within ProE. The user must manually determine the criticality of the parts within the model, whether they are suppressible or non-suppressible and whether the non-suppressible features are capable of simplification, based upon the analysis type, material, and the part feature. Simplification of the part can either be accomplished by the method proposed in the previous chapter or by replacing the complex part with a simpler geometrical part. After the criticality of the part is determined, the suppressible parts can be suppressed in groups, rather than one-by-one, based upon either their PART_NAME or MATERIAL value. This also takes into account the parent-child relationships developed within the model. To demonstrate the process, two different assembly models were simplified and analyzed. In comparison to manually simplifying the parts, the number of passes to suppress the non-critical parts was better than halved; while still increasing the mesh quality, reducing the mesh generation time, and acquiring analysis results within a given error of 5%.

Chapter 5

Conclusion

5.1 Intellectual Contributions

The contributions of this work can be separated into two primary categories:

1. A framework for simplifying CAD part models was developed with the utilities provided within ProE. The approach contains several key aspects. First, numerous features can be suppressed at one time, compared to the industry's current simplification process of suppressing one feature at a time. Second, a rule basis has been learned through expert demonstration with only conservative error, which can evolve as more parts are simplified. Third, the parent-child relationships were extracted from the model and used in the analysis to determine the suppressibility state of the model, thus, ensuring that a child feature was not unintentionally suppressed within the process. Finally, in order to reduce the potential effect of the simplification on the FEA results, the inter-feature distance to the boundary conditions were calculated to maintain any features within a close proximity of the boundary and force constraints.

2. A framework for simplifying CAD assembly models was developed with the utilities provided within ProE. The approach contains several key aspects. First, numerous parts can be suppressed at one time, compared to the indus-

try's current simplification process of suppressing one part at a time. Second, the parent-child relationships were extracted from the model and used when suppressing the parts, ensuring that a child part was not unintentionally suppressed within the process. Finally, integration of the part simplification was introduced within the assembly model simplification to simplify parts within the overall assembly model.

The work presented in this thesis provides a simplification tool for both part and assembly models, which is not the case for all model simplification tools.

5.2 Anticipated Benefits

The simplification frameworks can be extremely beneficial within the industry analysis. The overall analysis time for a model will likely be reduced, since the model preparation time can be reduced. In the part simplification approach discussed in Chapter 2, the process greatly reduces the users involvement in the process. Automatic identification and labeling of the non-critical features and multi-feature suppression of the identified non-critical features, upon the user's verification, will show a significant time reduction from the manual technique of the user analyzing each feature and suppressing them one at a time. In addition to reducing the model preparation time, errors involved in the simplifation part models may be reduced; especially in the accidental suppression of critical child features, since the the parent-child relationships are incorporated into the suppression algorithm. Reducing the errors within the analyses has two benefits. First it reduces the chance of passing

along erroneous results and the need to rerun an analysis because it was discovered to have suppressed a critical part.

In the assembly model simplification tool, numerous parts are capable of suppression compared to the industry version of suppressing part by part; reducing the model preparation time. Also, the integration of the part simplification tool into the assembly model simplification will reduce the time to simplify the parts within the overall model. Initially, these tools will not create a large impact in reducing the analysis time, but as time goes on there will be a gradual decrease in the model preparation time; since it will take time to gain the engineers trust.

5.3 Future Work

While this thesis has described successful frameworks for the simplification of both part and assembly CAD models for the use of downstream applications, such as FEA, there are many areas in which future work can offer further insight. The main areas of interest for future improvement are directly related to the work that has been complete in this thesis.

5.3.1 Improvement to the Part Simplification Framework

The part simplification framework presented in Chapter 2, but has several limitations. First, the framework is limited to suppressing only three types of features within the model; holes, rounds, and chamfers. Within the field, simplification of a part is not only limited to these three features. Expansion of the framework to in-

clude other features, for instance, bosses, slots, and steps, would greatly improve the robustness of the overall simplification tool. Another aspect of the framework that could be improved is the calculation of the feature distance to the boundary conditions. In the current framework the inter-feature distance calculation only takes into account the geometric distance. Within an analysis the topological distance and stress path can also have an effect within the criticality of the feature, so these should be incorporated into or included with the geometric distance calcultaion.

A third aspect for expansion is to incorporate the overall volumetric change between the original and simplified models. The overall volumetric change is similar to the parent-child relationship, in the fact that it may require features with parameters qualifying it to be a suppressible feature to be maintained within the model. For instance, one hole having parameters, which constitute the feature to be cionsidered non-critical, will not have an impact on the analysis results; although, taking this same hole and placing it numerous times throughout the model, generating a large volumetric change between the simplified and non-simplified model may affect the analysis results (even though the parameters of the holes constituted it to be classified as a non-critical feature).

5.3.2 Advancements in Assembly Simplification Approach

The assembly model simplification framework was more user intensive than that of the part simplification framework, allowing for more areas of work to be focused on in the future. First, the parent-child relationship would need to be

extended to allow for the inclusion of minor assembly models within the overall assembly model. If the parent part of an assembly model to be included is not used as the references for placement into the model, then they are technically a child in the major assembly, for which they are a parent to in the minor assembly model. Although this is the case, the possibility exists for minor assembly parts to become unreferenced in the overall model, if its child is used as the placement reference for the minor assembly and the suppressed for simplification in the overall model.

Another possible area for extension of the framework is to automatically determine the criticality; non-critical, semi-critical, or critical; since in the current framework the user must manually determine the criticality of each part. Determining the criticality of parts requires a knowledge of not only the part attributes, but also of the assembly attributes and analysis type. Chapter 4 illustrated that the same part within the same model can be replaced by a simpler geometry in one analysis type, but not within another analysis type. A third possible area for future work is automatic calculation and construction of a simpler part geometry for the parts capable of being replaced by the simpler geometry.

Bibliography

[1] Masters in the art of airbrush automotive illustration.

[2] A. and Sheffer. Model simplification for meshing using face clustering. *Computer-Aided Design*, 33(13):925 – 934, 2001.

[3] C. Andújar, P. Brunet, and D. Ayala. Topology-reducing surface simplification using a discrete solid representation. *ACM Transactions on Graphics (TOG)*, 21(2):88–105, 2002.

[4] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[5] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. *Stability and Computation of Medial Axes - a State-of-the-Art Report*. Mathematics and Visualization. Springer Berlin Heidelberg, 2009.

[6] DH Choi, TW Kim, and K. Lee. Multiresolutional representation of b-rep model using feature conversion. *Transactions of the Society of CAD/CAM Engineers*, 7(2):121–130, 2002.

[7] CS Chong, A. Senthil Kumar, and KH Lee. Automatic solid decomposition and reduction for non-manifold geometric model generation. *Computer-Aided Design*, 36(13):1357–1369, 2004.

[8] P. Dabke, V. Prabhakar, and S. Sheppard. Using features to support finite element idealizations. *Computers in Engineering*, 1:183–183, 1994.

[9] H. Date, S. Kanai, T. Kishinami, and I. Nishigaki. Flexible feature and resolution control of triangular meshes. In *Visualization, Imaging, and Image Processing*. ACTA Press, 2006.

[10] S. Dey, M. S. Shephard, and M. K. Georges. Elimination of the adverse effects of small model features by the local modification of automatically generated meshes. *Engineering with Computers*, 13:134–152, 1997. 10.1007/BF01221211.

[11] R. J. Donaghy, C. G. Armstrong, and M. A. Price. Dimensional reduction of surface models for analysis. *Engineering with Computers*, 16:24–35, 2000. 10.1007/s003660050034.

[12] B. Evans and D. Fisher. Overcoming process delays with decision tree induction. *IEEE Expert*, 9(1):60–66, 1994.

[13] M. Foskey, M.C. Lin, and D. Manocha. Efficient computation of a simplified medial axis. *Journal of Computing and Information Science in Engineering*, 3:274, 2003.

[14] S. Gao, W. Zhao, H. Lin, F. Yang, and X. Chen. Feature suppression based cad mesh model simplification. *Computer-Aided Design*, 42(12):1178–1188, 2010.

[15] A. Giordana, F. Neri, and L. Saitta. Automated learning for industrial diagnosis. *Fielded applications of machine learning. San Francisco: Morgan Kaufmann*, 1996.

[16] Blum H. A transformation for extracting new descriptors of shape. *Analysis and Machine Intelligence*, 17(1):635–640, 1967.

[17] S. Hari Gopalakrishnan and K. Suresh. Feature sensitivity: A generalization of topological sensitivity. *Finite Elements in Analysis and Design*, 44(11):696–704, 2008.

[18] Keisuke Inoue, Takayuki Itoh, Atsushi Yamada, Tomotake Furuhata, and Kenji Shimada. Face clustering of a large-scale cad model for surface mesh generation. *Computer-Aided Design*, 33(3):251 – 261, 2001.

[19] K. Jorabchi, J. Danczyk, and K. Suresh. Algebraic reduction of beams for cad-integrated analysis. *Computer-Aided Design*, 42(9):808–816, 2010.

[20] K. Jorabchi and K. Suresh. Nonlinear algebraic reduction for snap-fit simulation. *Journal of Mechanical Design*, 131:061004, 2009.

[21] N. Joshi and D. Dutta. Feature simplification techniques for freeform surface models. *Journal of Computing and Information Science in Engineering*, 3:177, 2003.

[22] Ronny Kohavi and J. Ross Quinlan. *Data mining tasks and methods: Classification: decision-tree discovery*, pages 267–276. Oxford University Press, Inc., New York, NY, USA, 2002.

[23] S. Koo and K. Lee. Wrap-around operation to make multi-resolution model of part and assembly. *Computers & Graphics*, 26(5):687–700, 2002.

[24] P. Langley and H.A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64, 1995.

[25] Jae Yeol Lee, Joo-Haeng Lee, Hyun Kim, and Hyung-Sun Kim. A cellular topology-based approach to generating progressive solid models from feature-centric models. *Computer-Aided Design*, 36(3):217 – 229, 2004.

[26] S.H. Lee. A cad–cae integration approach using feature-based multi-resolution and multi-abstraction modelling techniques. *Computer-Aided Design*, 37(9):941–955, 2005.

[27] S.H. Lee. Feature-based multiresolution modeling of solids. *ACM Transactions on Graphics (TOG)*, 24(4):1417–1441, 2005.

[28] S.H. Lee, K. Lee, and K.Y. Lee. Feature-based multiresolution and multi-abstraction non-manifold modeling system to provide integrated environment for design and analysis of injection molding products. In *Proceedings of the first Korea-China joint conference on geometric and visual computing*, 2005.

[29] Yong-Gu Lee and Kunwoo Lee. Geometric detail suppression by the fourier transform. *Computer-Aided Design*, 30(9):677 – 693, 1998.

[30] Ming Li and Shuming Gao. Estimating defeaturing-induced engineering analysis errors for arbitrary 3d features. *Computer-Aided Design*, 43(12):1587 – 1597, 2011.

[31] V. Mishra and K. Suresh. Efficient analysis of 3-d plates via algebraic reduction. ASME, 2009.

[32] J.R. Quinlan. Decision trees and decision-making. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):339–346, 1990.

[33] W.C. Regli and V.A. Cicirello. Managing digital libraries for computer-aided design. *Computer-Aided Design*, 32(2):119–132, 2000.

[34] M. Rezayat. Midsurface abstraction from 3d solid models: general theory and applications. *Computer-Aided Design*, 28(11):905–915, 1996.

[35] TT Robinson, CG Armstrong, and R. Fairey. Automated mixed dimensional modelling from 2d and 3d cad models. *Finite Elements in Analysis and Design*, 47(2):151–165, 2011.

[36] W.A. Samad and K. Suresh. Cad-integrated analysis of 3-d beams: a surface-integration approach. *Engineering with Computers*, pages 1–10, 2009.

[37] J.J. Shah and M. Mäntylä. *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. Wiley-interscience, 1995.

[38] Avneesh Sud, Mark Foskey, and Dinesh Manocha. Homotopy-preserving medial axis simplification. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, SPM '05, pages 39–50, New York, NY, USA, 2005. ACM.

[39] R. Sun, S. Gao, and W. Zhao. An approach to b-rep model simplification based on region suppression. *Computers & Graphics*, 34(5):556–564, 2010.

[40] Atul Thakur, Ashis Gopal Banerjee, and Satyandra K. Gupta. A survey of cad model simplification techniques for physics-based simulation applications. *Computer-Aided Design*, 41(2):65 – 80, 2009.

[41] S. Venkataraman and M. Sohoni. Reconstruction of feature volumes and feature suppression. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 60–71. ACM, 2002.

[42] H Zhu and C.H Menq. B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. *Computer-Aided Design*, 34(2):109 – 123, 2002.